DOCTORAL PROGRAMME IN
INFORMATION AND COMMUNICATION TECHNOLOGY

Doctoral candidate

Roberto Michele Tumolo

Cycle	**29°**
Thesis	**Title**
Advisor	**Paolo Rocca (University of Trento)**
Co-advisor	

1. List of publications

[R1] Giancarlo Prisco, Michele D'Urso, Roberto Michele Tumolo, "Maximally Sparse, Steerable and Non-superdirective Array Antennas Via Convex Optimizations", *IEEE Transactions on Antennas and Propagation*, vol. 64, Issue 9, pp. 3840–3849, September 2016.

[R2] R. M. Tumolo, M. D'Urso, G. Prisco, and A. Buonanno, "Fast synthesis of planar, maximally thinned arrays", *Progress In Electromagnetics Research Letters*, Vol. 68, 47-52, 2017.

2. Research/study activities

Research activities in the AESA (Active Electronically Scanned Array) ambit for the renewal of the Italian navy in Leonardo-Finmeccanica Company

UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE

ICT International Doctoral School

DESIGN, ANALYSIS, APPLICATION AND EXPERIMENTAL ASSESSMENT OF ALGORITHMS FOR THE SYNTHESIS OF MAXIMALLY SPARSE, PLANAR, NON-SUPERDIRECTIVE AND STEERABLE ARRAYS

Roberto Michele Tumolo

Advisor

Prof. Paolo Rocca

University of Trento

Tutors

Aniello Buonanno, Giancarlo Prisco, Michele D'Urso

Leonardo Finmeccanica Company

Abstract

This thesis deals with the problem of synthesizing planar, maximally sparse, steerable and non–superdirective array antennas by means of convex optimization algorithms and testing their performances on an existing array to assess its far field performances in terms of requirements fulfilment. The reason behind the choice of such topic is related to those applications wherein the power supply/consumption, the weight and the hardware/software complexity of the whole radiating system have a strong impact on the overall cost. On the other hand, the reduction of the number of elements has of course drawbacks as well (loss in directivity, which means a smaller radar coverage in radar applications, loss in robustness, etc.), however the developed algorithms can be utilized for finding acceptable trade-offs that arise, inevitably, when placing advantages and disadvantages of sparsification on the balance: it is only a matter of appropriately translating requirements in a convex way. The synthesis scheme will be described in detail in its generality at the beginning, showing how the proposed synthesis techniques outperform several results existing in literature and setting the bar for new benchmarks. In particular, an important, innovative constraint has been considered in the synthesis problem that prevents selection of elements at distances below half-wavelength: the non super-directivity. Moreover, an interesting result will be derived and discussed: the trend of the reduction of the number of elements Versus the (maximum) antenna size is decreasing as the latter increases. Afterwards the discussion will be focused on an existing antenna for radar applications, showing how the proposed algorithms intrinsically return a single layout that works jointly for transmitting and receiving (two-way synthesis). The results for the specific case chosen (mainly the set of weights and relative positions) are first numerically validated by a full-wave software (Ansys HFSS) and then experimentally assessed in anechoic chamber through measurements.

Keywords
Sparse array synthesis, convex optimization, numerical and on field evaluation

Contents

LIST OF TABLES IV

CHAPTER 1 1

INTRODUCTION 1
Thesis outline 3

CHAPTER 2 5

2.1 Introduction 5
2.2 Planar, regular arrays 5
2.3 Feasibility criteria 8
2.4 Planar array tapering 10
2.4.1 An example of array synthesis: Dolph-Chebyshev tapering 11
2.4.2 An example of array synthesis: Taylor tapering 14
2.5 Convex Optimization 19
2.5.1 Lines and segments 19
2.5.2 Affine sets 20
2.5.3 Convex sets 20
2.5.4 The convex hull 20
2.5.5 Cones 21
2.5.6 Norms and quasi-norms 21
2.5.7 Norm balls and norm cones 22

CHAPTER 3 23

3.1 Introduction 23
3.2 The Optimum SLL 23
3.3 Problem statement: the minimization of the ℓ_0 pseudo-norm 25
3.3.1 THE STEERABILITY 27
3.3.2 UPPER AND LOWER BOUNDS 27
3.3.3 TWO-WAYS SYNTHESIS 28
3.3.4 THE NON SUPER-DIRECTIVITY MATRIX 28
3.3.5 THE SHARED LAYOUT 31
3.3.6 THE COST FUNCTION 31
3.4 Problem Solution: reweighted ℓ_1-norm and B&B algorithms 34
3.5 B&B algorithms 35
- VARIABLE WITH FRACTIONAL VALUE CLOSEST TO 0.5 36
- VARIABLE WITH HIGHEST IMPACT ON THE OBJECTIVE FUNCTION 36
- VARIABLE WITH THE LEAST INDEX 36

3.5.1 An example of IP problem 37
3.5.2 An example of IP problem with applied criteria 40
3.6 Reweighted ℓ_1-norm 43
3.7 The dynamics 45

CHAPTER 4 46

4.1 Introduction 46
4.2 Comparison with literature Benchmarks 47
4.3 New benchmarks 57

CHAPTER 5 61

5.1 Introduction 61
5.2 The X-antenna: features and requirements 62
5.2.1 THE TX REQUIREMENTS 62
5.2.2 THE RX REQUIREMENTS 62
5.3 Comparison between simulated results: the Tx Synthesis 63
5.4 Comparison between simulated results: the Rx Synthesis 69
5.5 Comparison between simulated and measured results 73
5.6 Experimental results: the measured FF in Tx of the sparse X-Antenna 75
5.7 Experimental results: the measured FF in Rx of the sparse X-Antenna 77

CHAPTER 6 81

CONCLUSIONS AND FUTURE RESEARCH 81

Published Journal Papers

[R3] Giancarlo Prisco, Michele D'Urso, Roberto Michele Tumolo, "Maximally Sparse, Steerable and Non-superdirective Array Antennas Via Convex Optimizations", *IEEE Transactions on Antennas and Propagation*, vol. 64, Issue 9, pp. 3840–3849, September 2016.

[R4] R. M. Tumolo, M. D'Urso, G. Prisco, and A. Buonanno, "Fast synthesis of planar, maximally thinned arrays," *Progress In Electromagnetics Research Letters*, Vol. 68, 47-52, 2017.

List of Tables

Table 1. Array SLL Versus edge illumination parameter α _____ *15*
Table 2. Coordinates and normalized weights for the array in Figure 31. _____ *49*
Table 3. Coordinates and normalized weights of the flat-top in Figure 35. _____ *51*
Table 4. Input parameters: array edge, BWs and SLLs _____ *57*

List of Figures

Figure 1. Generalized array configuration _____ 6
Figure 2. Geometry of the considered planar rectangular grid. _____ 7
Figure 3. Chebyshev normalized 2D power pattern. _____ 12
Figure 4. Normalized Chebyshev tapering. _____ 13
Figure 5. Horizontal cut of a planar Chebyshev tapering. In the figure equiripple SLLs at -40dB are clearly visible _ 13
Figure 6. Vertical cut of a planar Chebyshev tapering. In the figure equiripple SLLs at -30dB are clearly visible ___ 14
Figure 7. Taylor normalized 2D power pattern. _____ 17
Figure 8. Normalized Taylor tapering. _____ 17
Figure 9. Horizontal cut of a planar Taylor tapering. In the figure it can be seen that the first 5 sidelobes are Chebyshev-like ($\bar{n}=5$), whereas from the sixth one a sinc-type decay begins _____ 18
Figure 10. Vertical cut of a planar Taylor tapering. In the figure it can be seen that the first 3 sidelobes are Chebyshev-like ($\bar{n}=3$), whereas from the fourth one a sinc-type decay begins _____ 18
Figure 11. The line passing through x_1 and x_2 is described parametrically by $\vartheta\mathbf{x}_1+(1-\vartheta)\mathbf{x}_2$, where ϑ varies over $\mathbb{R}$. The line segment between x_1 and x_2, which corresponds to ϑ between 0 and 1, is shown darker. _____ 19
Figure 12. The convex hulls of two sets in $\mathbb{R}^2$. Left. The convex hull of a set of fifteen points (shown as dots) is the pentagon that encompasses them. Right. The convex hull of a non convex set. _____ 20
Figure 13. The pie slice shows all points of the form $\vartheta_1\mathbf{x}_1+\vartheta_2\mathbf{x}_2$, where $\vartheta_1,\vartheta_2\geq 0$. The apex of the slice (which corresponds to $\vartheta_1=\vartheta_2=0$) is at 0; its edges (which correspond to $\vartheta_1=0$ or $\vartheta_2=0$) pass through the points x_1 and x_2 . _____ 21
Figure 14. Boundary of second-order cone in R^3 _____ 22
Figure 15. Horizontal cuts of a 3λ square array with rectangular lattice in the tapered (blu line) and isophoric (red) cases. The optimum SLL is -22.9784dB [49]. _____ 24
Figure 16. Tapering of a 3λ square array with optimum SLL equal to -22.9784dB [49]. _____ 25
Figure 17. 7λ spectrum (the colourbar is set in such way that only the main lobe is visible). The desired steering is 57° either in azimuth and elevation, so that the ellipse beyond the visible circle is in this case a circle as well of ray 1+sin(57°)≈1.83 (dotted white circle). _____ 27
Figure 18. Single column of matrix A. The red points represent all the points at distance smaller than half-wavelength with respect to the circle's center, whereas the blue ones represent all the points at distance greater than or equal half-wavelength. _____ 29
Figure 19. Single column of matrix A. The red points represent all the points at distance smaller than half-wavelength with respect to the circle's center, whereas the blue ones represent all the points at distance greater than or equal half-wavelength _____ 30
Figure 20. Example of four circles, related to four specific columns of A, on a general array layout identified by $\tilde{\mathbf{w}}_b$. _____ 30
Figure 21. Unit balls in R^2 for the l_p quasi-norm for various values of p. Note that as p→0, the unit ball collapses onto the (x,y) axes (the case p=0, highlighted in red, is the binary case "x or y"). _____ 32
Figure 22. Example of convex geometry: unit ball in R^3 for the l_p norm for p=3. _____ 33
Figure 23. Example of concave geometry: unit ball in R^3 for the l_p quasi-norm for p=0.5. _____ 34
Figure 24. LP and IP feasible regions _____ 38
Figure 25. LP and IP feasible regions after the first branching. _____ 39
Figure 26. LP and IP solution regions after the second branching. _____ 39
Figure 27. B&B tree for problem (2.15). _____ 40
Figure 28. B&B tree for problem (2.16) using best-bound-first _____ 41
Figure 29. B&B tree for problem (2.16) using depth-first _____ 42
Figure 30. ℓ_0-like functions. At the origin, the canonical ℓ_0 sparsity count $f_0(t)$ is better approximated by the log-sum penalty function $f_{log,\varepsilon}(t)$ than by the traditional convex ℓ_1 relaxation $f_1(t)$. _____ 45
Figure 31. Resulting array layout relative to the synthesis of focused beam power pattern. Active radiating elements N_e=35. Discretization grid: $d_x=d_y=0.1\lambda$. The dashed curve represents the convex hull. _____ 47
Figure 32. 3D normalized power pattern radiated by the array in Figure 31 with N_e=35 active radiating elements. BW= $\sin(\vartheta_{bw})$=0.240 at -6dB. SLL=-17.5dB. _____ 48
Figure 33. Normalized magnitude in [dB] of the resulting pattern radiated by the array in Figure 31. The visible region is delimited by the withe circle. N_e=35 active radiating elements. _____ 48
Figure 34. ϕ-cutting planes of the normalized power pattern radiated by the array in Figure 31 with N_e=35 active radiating elements. _____ 49

Figure 35. Resulting array layout relative to the synthesis of the flat-top power pattern. Actual radiating elements Ne=57. Discretization grid: dx=dy=0.01λ. The dashed curve represents the convex hull. 50
Figure 36. 3D normalized flat-top power pattern radiated by the array in Figure 35 with N_e=57 elements. SLL=-24.53dB. 50
Figure 37. Normalized power in [dB] of the resulting flat-top pattern in the (u,v) domain. The visible region is delimited by the withe circle, Ne=57 active radiating elements. 51
Figure 38. ϕ-cutting planes of the normalized flat-top power pattern synthesized in [81] (dotted lines) and the one synthesized with the proposed synthesis scheme (solid lines). Ne=57 active radiating elements. 51
Figure 39. Array of 150 elements obtained in [51] 53
Figure 40. Array of 119 elements obtained by means of our proposed algorithms 53
Figure 41. Circular shape 3D pattern 54
Figure 42. 2D circular pattern. The two white circles represent the mainlobe and SL boundaries respectively 54
Figure 43. 2D circular pattern cuts. The vertical (u = 0) and horizontal (v = 0) cuts are reported in the figure together with the boundary required by the benchmark. Datatips have been inserted to show precisely the SLL. 55
Figure 44. Diamond-shape 3D pattern 55
Figure 45. 2D diamond-shape pattern. The white circle and dashed square, together with the red dash-dotted square represent the null region, the mainlobe and sidelobe regions boundaries respectively. 56
Figure 46. 2D diamond-shape cuts. The vertical (u = 0) and horizontal (v = 0) cuts are reported in the figure together with the boundary required by the benchmark. Datatips have been inserted to show precisely the SLL. 56
Figure 47. ENRR Vs steering angles for BWs and SLLs of Table 4. 57
Figure 48. Resulting steerable array relative to the synthesis of pencil beam power pattern. Active radiating elements N_e=163, edge size 7λ, steering range 57°. The dash-dotted curve represents its convex hull 58
Figure 49. 3D power pattern radiated by the 7λ steerable array (Figure 48) pointing at broadside. Ne=163 active radiating elements. BW=18.57°, SLL≅-25dB 59
Figure 50. ϕ-cutting planes for ϕ=0°, 30°, 45° and 90° of the normalized power pattern of Figure 49. On the abscissa it is $r \in [-1-\sin(57°), 1+\sin(57°)]$ 59
Figure 51. 2D power pattern of the 7λ steerable array depicted in Figure 48 pointing at $(u_0,v_0)=[\sin(57°),0]$ *and with the input parameters in Table 4. The visible region is delimited by the white circle, whereas the dashed one identifies the region* Φ 60
Figure 52. A photograph of the X-antenna. 61
Figure 53. The full X-Antenna Tx layout 64
Figure 54. The sparse X-Antenna Tx layout 65
Figure 55. Horizontal cut for sparse Tx X-Antenna at broadside 66
Figure 56. Vertical cut for sparse Tx X-Antenna at broadside. 66
Figure 57. 45° azimuthal (v=0) cut for sparse Tx X-Antenna. 67
Figure 58. -45° azimuthal (v=0) cut for sparse Tx X-Antenna. 67
Figure 59. 60° elevation (u=0) cut for sparse Tx X-Antenna. 68
Figure 60. -60° elevation (u=0) cut for sparse Tx X-Antenna. 68
Figure 61. The sparse X-Antenna Rx layout 69
Figure 62. Vertical cut for sparse Rx X-Antenna at broadside 70
Figure 63. Horizontal cut for sparse Rx X-Antenna at broadside. 70
Figure 64. 45° azimuthal (v=0) cut for sparse Rx X-Antenna. 71
Figure 65. -45° azimuthal (v=0) cut for sparse Rx X-Antenna. 71
Figure 66. 60° elevation (u=0) cut for sparse Rx X-Antenna. 72
Figure 67. -60° elevation (u=0) cut for sparse Rx X-Antenna 72
Figure 68. Vertical cut for sparse Tx X-Antenna at broadside: comparison between measured and simulated (HFSS) results 73
Figure 69. Horizontal cut for sparse Tx X-Antenna at broadside: comparison between measured and simulated (HFSS) results 74
Figure 70. 45° azimuthal (v=0) cut for sparse Tx X-Antenna: comparison between measured and simulated (HFSS) results 74
Figure 71. Vertical cut for sparse Tx X-Antenna at broadside 75
Figure 72. Horizontal cut for sparse Tx X-Antenna at broadside 76
Figure 73. 45° azimuthal (v=0) cut for sparse Tx X-Antenna 76
Figure 74. 60° elevation (u=0) cut for sparse Tx X-Antenna 77
Figure 75. Vertical cut for sparse Rx X-Antenna at broadside 78
Figure 76. Horizontal cut for sparse Rx X-Antenna at broadside 79
Figure 77. 45° azimuthal (v=0) cut for sparse Rx X-Antenna 79

Figure 78. 60° elevation (u=0) cut for sparse Rx X-Antenna _____ *80*
Figure 79. 60° elevation (u=0) cut for sparse and full Rx X-Antenna _____ *80*

Chapter 1

Introduction

The design of sparse Active Electronically Scanned Arrays (AESAs) radiating a desired Far Field (FF) pattern, satisfying a prescribed power mask with as few elements as possible has several potential applications like satellites, radars, etc., as well as interests on a theory basis. The synthesis of unequally spaced arrays is in general a complex problem that cannot be properly addressed with analytical methods, thus mainly leaving the task to numerical techniques.
Deterministic approaches have been proposed in [1-4], where simple formula are derived for synthesizing given FF patterns by using uniform amplitude sparse arrays. In [1], [2] and [3] in particular, a large number of sensors is often required to fulfill the design constraints. Furthermore, the number of sensors is an input of the solution methods and it is not derived once the FF performances to achieve have been given. A novel deterministic, non-iterative synthesis algorithm based on the matrix pencil method has been proposed in [40-[41]. It efficiently reduces the number of elements in a linear antenna array with very short computation time. Such approaches allow to determine the minimum number of sensors to fit, with a fixed accuracy, provided that a proper reference pattern is given in amplitude and phase as input.

On the other hand, existing global optimization methods like Genetic Algorithm (GA) [5-17], Simulated Annealing (SA) [17-23], Differential Evolution (DE) [24-31] and Particle Swarm Optimization (PSO) [32-39] can be used for the synthesis of non-uniform linear and planar arrays, although such algorithms, especially the GA, require a computational load that can be likely prohibitive for arrays with large number of elements and constraints, and besides they do not assure global solutions.
In general, of the different approaches present in literature on this topic, that can be grouped according to specific criteria, one in particular is of interest for the present work: the distinction between pattern-fitting problems and mask-constrained ones [42]. In the first case, which, to the student's best knowledge, is the most taken under consideration in literature, it is worth noting that the field distribution satisfying the given design constraints is not unique and different choices for the reference field pattern lead to different results, in terms of number of sensors. Therefore there is no guarantee that the number of antennas of the resulting sparse array is the minimum one. Such problem represents one of the main drawbacks for many existing approaches based on Compressed Sensing [42] [76] and Bayesian CS (BCS) [42] [45], wherein the reference field to fit in order to obtain the maximally sparse array is a priori assumed known or anyway fixed a priori. Resorting to pattern-fitting approaches in the radar field for example is quite constraining to the best of the student's knowledge, since it dramatically decreases the available degrees of freedom. In fact, most of the requirements are derived from Probability of Detection (PD), Probability of False Alarms (PFA) and range coverage demands, which, in terms of synthesis goals, only define upper and lower bounds to be met, not a precise field pattern. Hence, as the main application selected in this work is AESAs for surveillance purposes (radars), and since the whole PHD has been carried on in collaboration with the *University of Trento* while working as a *Leonardo – Finmeccanica* employee, an emphasis will be given to the second case, *i.e.* mask-constrained synthesis [46] [50] [77] by using convex optimization techniques [55], [56], [58].

One of the frameworks the sparse array synthesis can be set in is CS [42-43]. Roughly speaking indeed, the measurements one would use in the recovery of a sparse signal are substituted by FF specifications, while the array layout is actually the "sparse signal" to be recovered. In particular, the aim is generally to fit 'at best' a reference complex pattern, and the problem's actual unknowns are both the elements excitations and locations. A very dense grid is usually chosen for the array so that the elements whose excitation is different from zero also determine the actual locations to be used. However, one of the main drawbacks of existing solutions available in literature [46], [50] is that superdirectivity (according to the definition given by Hansen [51] <<A*n array is superdirective if its directivity is higher than the one obtained from the isophoric case (uniform excitations with constant amplitude and linear phase)*>>) has not

been prevented: arrays with distances below half-wavelength are obtained after optimization. Nevertheless, some attempts at clustering elements at distances below half-wavelength have been made in literature [52-54] by setting constraints on the physical size of the antenna. Anyway, such methods are essentially applied in post-processing, so that no a priori constraint on the inter-element distance is fixed in the synthesis routine. This is one of the main reasons why in this entire work particular attention has been paid to the non-superdirectivity and thus to the methods that allow to achieve it a priori in the synthesis process. Furthermore, imposing a minimum inter-element distance equal to half-wavelength in the array resulting from the synthesis makes the problem of mutual coupling [74] [75] among the radiating elements negligible. As it will be described in detail in the following chapters, preventing superdirectivity a priori in an exact way (as we did), as well as considering the sparsification process in terms of switching on and off some elements, requires the introduction of binary variables (more in general of the ℓ_0-pseudo-norm), whose management falls into the context of Mixed Integer Programming (MIP) [57] [59] [61]. Solution of the synthesis minimization problem in this picture is efficiently achieved by employing techniques like Fast Branch&Bound (FB&B), which are tailored for MIP problems. The B&B algorithm [57], [66] is a general method for finding the optimal solution to problems of combinatorial complexity (NP-hard) that enumerates candidate solutions by means of a rooted-tree space search. According to its implementation and to the upper and lower bounds of the cost function, it can prune huge amounts of branches, thus massively shortening the global solution search. Besides, it can be efficiently implemented with a parallel approach for optimizing the simultaneous use of cores and machines [69].

These techniques were firstly adopted by Narendra and Fukunaga [62], allowing several improvements (in computational cost saving) later on [63], [67] [68]. From a technical point of view, we have decided to utilize B&B (historically employed in mixed integer problems, [64] [65]) since the cost function we are dealing with is intrinsically upper and lower bounded (as it will be seen in the later chapters), which prunes *ab initio* several branches of the span tree. In some cases global optimization schemes like the GA are exploited in MIP problems and can aid the B&B based optimization [70], in others the SA is used [71] [72]; in general, exact solution strategies and heuristic methods can be used to solve this class of problems. The B&B is one of the most common ones and gives an exact solution, whereas algorithms like GA and SA represent non-exact approaches in the general purpose heuristic methods. Moreover, by using algorithms like GA, SA and DE we experienced much longer simulation time even for much smaller-scaled problems (linear arrays for instance, [47]) than the ones carried on and discussed in the present work. For the current goals, they become computationally prohibitive. Anyway, whether exact or non-exact approaches perform better for a given problem depends on how they are implemented. Since the B&B explores potentially the huge tree given by the combinatorial problem, it is all a matter of how many branches of the search tree are pruned and how quickly.

Another way to solve, approximately, the combinatorial problem that directly derives from the purpose of switching on and off the elements of the array with a constraint on their minimum distance is to use the reweighted ℓ_1-norm [73]. In fact, from the CS theory we know it is possible to reconstruct sparse signals exactly from what appear to be highly incomplete sets of linear measurements by constrained ℓ_1 minimization in the sense that substantially fewer measurements are needed for exact recovery. In view of the aforementioned way of considering the array layout in terms of “sparse signal” to recover, in [73] the algorithm consists of solving a sequence of weighted ℓ_1-minimization problems where the weights used for the next iteration are computed from the value of the current solution.

Since either B&B and the reweighted ℓ_1-norm have been selected for the optimization scheme discussed in this thesis, the trade-off between them will be discussed in detail subsequently, as well as a generic criterium for choosing the former or the latter for the goal at hand.

Starting from the basic scheme of [78] and [47], in this thesis work a new synthesis method for achieving maximally sparse**,** planar, steerable, non-superdirective arrays is proposed, and the problem of deriving the trend of the reduction in the number of radiating elements *Versus* either steering intervals and array size is also addressed [48]. In the proposed synthesis scheme, given a desired power mask and the array geometry, it is first derived the minimum array support containing a non-superdirective source whose radiation pattern fulfills the design constraints by exploiting the results in [79].

Then the maximally sparse, non-superdirective array, able to radiate the steerable power pattern satisfying the prescribed power mask is synthesized. Several literature benchmarks have been chosen [46] [80-81] in order to prove the effectiveness of the developed algorithms (even for shared-layout multi-pattern solutions [81]), and new benchmarks have been set [48].

In the process of sparsifying the AESA antenna chosen as final application of the developed algorithms, the element factor and all its pre-existing requirements have been taken into account, showing an convincing coherence with the results of the full-wave software Ansys HFSS and the anechoic chamber measurements, thus eventually and definitely proving the potential for future AESA projects.

Thesis outline

The thesis is structured as follows: chapter 2 introduces the mathematical background necessary for discussing the proposed synthesis schemes described in chapter 3. In the latter the mathematical problem is formulated and the developed solutions presented. Chapter 4 deals with the application of such solutions, in particular to either benchmarks that exist in literature and new ones, showing how they perform. Finally, chapter 5 shows the results of the application of the developed algorithms to an existing antenna, showing how its sparse version performs in FF (in some cases also with respect to the full, standard one). Conclusions end the work.

Chapter 2

Mathematical background

In this chapter the basics for the discussion in the next chapters are introduced and described. In particular, the conditions for the a priori existence of a power pattern lying within a given mask by providing shape and size of the pattern source are furnished. Then the planar array tapering is presented and two cases discussed: Chebyshev and Taylor set of weights.
Finally, all the definitions that gravitate around the convex optimization scheme and that are necessary for the description of the developed algorithms are provided.

2.1 Introduction

The introduction in this chapter of the fundamental equations that lay the basis of the (planar) array synthesis is preliminary to the next chapters and to the methods adopted throughout the PHD study. Specifically, the equations that describe the array factor and the element factor are reported [82] in order to show how and where the set of weights, that are fundamental to the synthesis process, come into play. Next, resorting to the works of Bucci, Isernia and Fiorentino [83], a simple and effective necessary condition to test if a source of given size and structure can radiate or not a power pattern lying in a given mask is provided. It is also briefly described in which cases the criterion is also sufficient. Then a synthetic discussion on planar array tapering is presented by referring to the general formulation according to Shelkunov [89] with two special cases: Dolph-Chebyshev [88] and Taylor [90]. In particular, the former has been taken into account because it is the type of array tapering that has been chosen for the general formulation of the developed algorithms for sparse synthesis. Eventually, the basic definitions of convex optimization, convex and affine sets, cones and convex hulls are given [55] and will be recalled when discussing the structure of the developed algorithms and one of the figures of merit for assessing the synthesized array performances (the convex hull).

2.2 Planar, regular arrays

Let us make a brief mention of the classical theory of planar arrays [82], since they represent the fundamental brick for building the entire mathematical formulation.

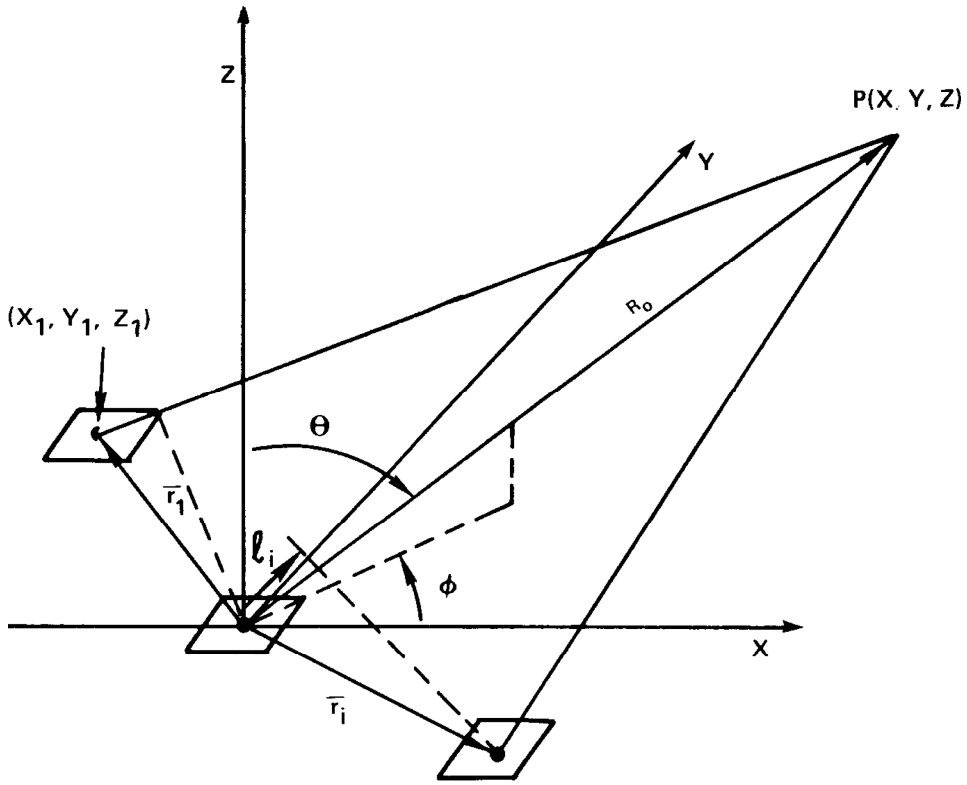

Figure 1. Generalized array configuration

The sketch in Figure 1 portrays a generalized distribution of array elements, here shown as small radiating surfaces. Each element radiates a vector directional pattern that has both angle and radial dependence near the element. However, for distances very far from the element, the radiation has the $[\exp(-jkR)]/R$ dependence of a spherical wave multiplied by a vector function of angle $\mathbf{f}_i(\vartheta,\phi)$ called the element pattern. Although this vector function depends on the kind of element used, the FF of any ith element can be written as

$$\mathbf{E}_i(r,\vartheta,\phi) = \mathbf{f}_i(\vartheta,\phi)e^{-\frac{jkR_i}{R_i}} \tag{1.1}$$

for

$$R_i = \sqrt{\left[\left(x-x_i\right)^2 + \left(y-y_i\right)^2 + \left(z-z_i\right)^2\right]} \tag{1.2},$$

where $k = \dfrac{2\pi}{\lambda}$ is the free-space wave number at frequency *f*. If the pattern is measured at a distance very far from the array (FF), then the above exponential can be approximated by reference to a distance *R* measured from an arbitrary center of the coordinate system. Since $R_i \approx R - \hat{\mathbf{r}}\cdot\mathbf{r}_i \approx R$, for $kR_i \rightarrow \infty$, where

$$\begin{aligned} &\mathbf{r}_i = \hat{\mathbf{x}}x_i + \hat{\mathbf{y}}y_i + \hat{\mathbf{z}}z_i \\ &\hat{\mathbf{r}} = \hat{\mathbf{x}}u + \hat{\mathbf{y}}v + \hat{\mathbf{z}}w \\ &(u,v,w) = (\sin\vartheta\cos\phi, \sin\vartheta\sin\phi, \cos\vartheta) \end{aligned} \tag{1.3},$$

then

$$e^{-\frac{jkR_i}{R_i}} = e^{-\frac{jkR}{R}} e^{jk\hat{\mathbf{r}}\cdot\mathbf{r}_i} \tag{1.4}$$

The required distance R for which one can safely use the FF approximation depends on the degree of fine structure desired in the pattern. Using the distance

$$R = 2\frac{D^2}{\lambda} \tag{1.5},$$

in which D is the largest array dimension, is adequate for most pattern measurements. With reference to Figure 2, for an arbitrary planar, rectangular grid composed of N x M sources, one can generally write the pattern by superposition:

$$\mathbf{E}(r,\vartheta,\phi) = \frac{e^{(-jkR)}}{R}\sum_{n=1}^{N}\sum_{m=1}^{M}\mathbf{f}_{nm}(\vartheta,\phi)w_{nm}\,e^{jk\hat{\mathbf{r}}\cdot\mathbf{r}_{nm}} \tag{1.6}$$

where w_{nm} are the array weights (voltages or currents) in the general case (complex weights), and we set $\mathbf{r}_{nm} = \hat{\mathbf{x}}x_n + \hat{\mathbf{y}}y_m$ and $z_i = 0$ for convenience.

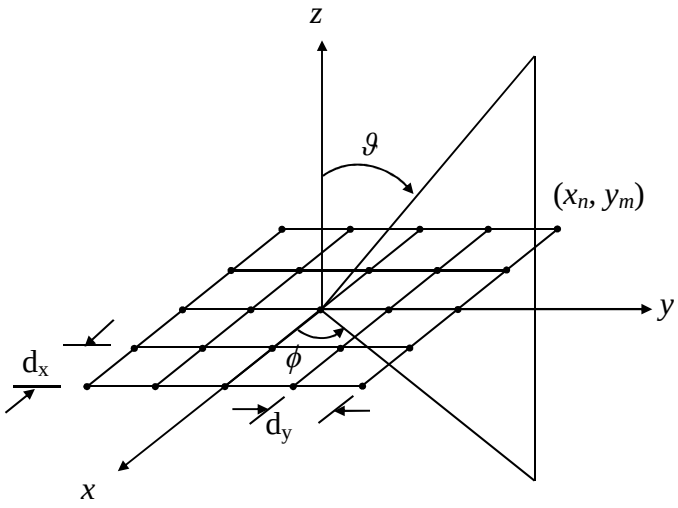

Figure 2. Geometry of the considered planar rectangular grid.

In general, the vector element patterns are different for each element in the array, even in an array of like elements; the difference is usually due to the interaction between elements near the array edge. However, it will be assumed throughout this thesis work that all patterns in a given array are the same. In this case, (1.6) becomes

$$\mathbf{E}(r,\vartheta,\phi) = \frac{e^{(-jkR)}}{R}\mathbf{f}(\vartheta,\phi)\sum_{n=1}^{N}\sum_{m=1}^{M}w_{nm}\,e^{jk\hat{\mathbf{r}}\cdot\mathbf{r}_{mn}} \tag{1.7}.$$

It is customary to remove the factor $[\exp(-jkR)]/R$ because the pattern is usually described or measured on a sphere of constant radius and this factor is just a normalizing constant. Thus, one can think of the pattern as being the product of a vector element pattern $\mathbf{f}(\vartheta,\phi)$ and a scalar function, called the Array Factor (AF):

$$\mathbf{E}(\vartheta,\phi) = \mathbf{f}(\vartheta,\phi)\cdot AF(\vartheta,\phi) \tag{1.8}$$

where

$$AF(\vartheta,\phi) = \sum_{n=1}^{N}\sum_{m=1}^{M} w_{nm}\, e^{jk\hat{\mathbf{r}}\cdot\mathbf{r}_{mn}} \tag{1.9}$$

and

$$\begin{aligned} &\hat{\mathbf{r}}\cdot\mathbf{r}_{mn} = nd_x u + md_y v \\ &k = k_0 = \frac{2\pi}{\lambda_0} \end{aligned} \tag{1.10}$$

meaning, in the second relation of (1.10), that we are considering a single frequency. Let us modify a bit the first relation of (1.10) by adding a couple (u_0, v_0) that represents the steering direction along which the array points:

$$AF(\vartheta,\phi) = \sum_{n=1}^{N}\sum_{m=1}^{M} w_{nm}\, e^{jk_0\left[nd_x(u-u_0)+md_y(v-v_0)\right]} \tag{1.11}$$

2.3 Feasibility criteria

Before going into details of the convex optimization schemes developed during the PHD an important role is played by the necessary (and possibly sufficient) conditions for the a priori existence of a power pattern lying within a given mask by providing shape and size of the pattern source. The problem is shown to be equivalent to establish if a system of linear inequalities admits a solution [83]; in particular, the criterion is both necessary and sufficient in the case of linear and uniform arrays, whereas it is only sufficient in the case of planar arrays with 2D factorable masks. In all practical instances, electromagnetic fields can be considered as belonging to a finite dimensional space. This happens not only when the source has by itself a finite number of degrees of freedom, such as for instance, in the case of an array antenna, but also when arbitrary radiating systems are considered. In the general case the number of degrees of freedom of the field is defined as the minimum number of independent parameters required for its representation within a given accuracy. It turns out that fields radiated from sources of bounded energy enclosed in a sphere of radius *a* can be effectively (i.e., in an efficient and not redundant way) approximated with bandlimited functions of bandwidth slightly larger than $k_0 a$ [84]. Accordingly, each FF component can be represented in a sampling series [85]:

$$\begin{aligned} E(\vartheta,\phi) \cong\, & E(0,\phi)\,\mathrm{D}_M(\vartheta) + \sum_{n=1}^{M}\Big[D_M(\vartheta-\vartheta_n)\sum_{m=-M_n}^{M_n} E(\vartheta_n,\phi_{n,m})D_{M_n}(\phi-\phi_{n,m}) \\ & -D_M(\vartheta+\vartheta_n)\sum_{m=-M_n}^{M_n} E(\vartheta_n,\phi_{n,m})D_{M_n}(\phi+\pi-\phi_{n,m})\Big] \end{aligned} \tag{1.12}$$

where

$$\vartheta_n = \frac{2\pi n}{2M+1},\ \phi_{n,m} = \frac{2\pi m}{2M+1},\ M \ge k_0 a,\ M_n \ge k_0 a \sin\vartheta_n$$

and

$$D_M(x) = \frac{\sin\left(\frac{2M+1}{2}x\right)}{(2M+1)\sin\left(\frac{x}{2}\right)} \tag{1.13}$$

is the Dirichlet sampling function. As a consequence, squared amplitude distributions can be represented by bandlimited functions [83], so that we can write:

$$\begin{aligned} \left|E(\vartheta,\phi)\right|^2 = P(\vartheta,\phi) \cong P(0,\phi)\,\mathrm{D}_M(\vartheta) + \sum_{n=1}^{2M}[D_{2M}(\vartheta-\vartheta_n)\sum_{m=-2M_n}^{2M_n}\mathrm{P}(\vartheta_n,\phi_{n,m})D_{2M_n}(\phi-\phi_{n,m}) \\ -D_{2M}(\vartheta+\vartheta_n)\sum_{m=-2M_n}^{2M_n}\mathrm{P}(\vartheta_n,\phi_{n,m})D_{2M_n}(\phi+\pi-\phi_{n,m})] \end{aligned} \tag{1.14}$$

where

$$\vartheta_n = \frac{2\pi n}{4M+1},\ \phi_{n,m} = \frac{2\pi m}{4M_n+1} \tag{1.15}$$

and $P(\vartheta_n,\phi_{n,m})$ denotes the corresponding sample of $\left|E(\vartheta_n,\phi_{n,m})\right|^2$. For planar equispaced arrays we have:

$$\begin{aligned} P(u,v) = \mathrm{c}_{00} + \sum_{q=1}^{M-1}\left[c_{0q}\cos(qk_0d_y v) + s_{0q}\sin(qk_0d_y v)\right] + \sum_{p=1}^{N-1}[c_{p0}\cos(\mathrm{p}\,k_0d_x\,\mathrm{u}) + \\ + s_{p0}\sin(\mathrm{p}\,k_0d_x\,\mathrm{u})] + \sum_{p=1}^{N-1}\sum_{\substack{q=-M+1 \\ q\neq 0}}^{M-1}[c_{pq}\cos(\mathrm{p}\,k_0d_x\,\mathrm{u} + qk_0d_y v) + \\ + s_{pq}\sin(\mathrm{p}\,k_0d_x\,\mathrm{u} + qk_0d_y v)] = \sum_{p=1-N}^{N-1}\sum_{q=1-M}^{M-1}P_{pq}D_{2N-1}(k_0d_x\,\mathrm{u} - \mathrm{u}_p)D_{2M-1}(k_0d_y\,\mathrm{v} - v_q) \end{aligned} \tag{1.16}$$

where

$$u_p = \frac{2\pi}{2N-1}\mathrm{p},\ v_q = \frac{2\pi}{2M-1}q,\ P_{pq} = P(u_p,v_q)\,.$$

The above results can be summarized introducing the generic finite dimensional representation:

$$P(\vartheta,\phi) = \sum_{p=1}^{T} D_p\Psi_p(\vartheta,\phi) \tag{1.17}$$

which, by appropriate choice of T and $\Psi_p(\vartheta,\phi)$, is representative of (1.16). It must be noted [83] that not all functions expressible as in (1.17) correspond to squared amplitude distributions. This is because T is larger than the number of (real) degrees of freedom of the field. Accordingly, the set of all squared amplitude distributions is a (non linear) variety embedded in the space spanned by the $\Psi_p(\vartheta,\phi)$ functions. However, (1.17) provides the smallest linear space containing the set of all squared amplitude distributions [83]. Exploitation of (1.17) makes it easy to show which conditions must be fulfilled so that a pattern lying in a given mask can exist. In fact, as (1.17), by construction, is able to represent all possible patterns radiated from a given classes of sources, a necessary condition for the existence of a field fulfilling given constraints is that the following system of linear inequalities in the variables D_p is satisfied:

$$\begin{cases} \sum_{p=1}^{T} D_p \Psi_p(\vartheta,\phi) \le \mathrm{UB}(\vartheta,\phi) \\ \sum_{p=1}^{T} D_p \Psi_p(\vartheta,\phi) \ge L\,\mathrm{B}(\vartheta,\phi) \end{cases} \tag{1.18}$$

By taking into account the bandlimitedness of $P(\vartheta,\phi)$ inequalities (1.18) can be substituted by a sufficiently fine discretization, so that

$$\begin{cases} \sum_{p=1}^{T} D_p \Psi_p(\vartheta_i,\phi_j) \le \mathrm{UB}(\vartheta_i,\phi_j) \\ \sum_{p=1}^{T} D_p \Psi_p(\vartheta_i,\phi_j) \ge L\,\mathrm{B}(\vartheta_i,\phi_j) \end{cases} \tag{1.19}$$

$$for\ i = 1,\ldots,L_1,\ \ j = 1,\ldots,L_2$$

is a system of ordinary linear inequalities in the variable D_p. The solvability of a system of linear inequalities is a well known problem, and it is equivalent to assess the existence of a "feasible point" for a linear programming problem [86].

The question amounts to establish, once a solution satisfying (1.19) has been obtained, if one can effectively get a field corresponding to that solution. Because, as it has been hinted before, $T \ge 2C$, wherein C is the number of complex degrees of freedom of the field [87], the set of (mathematically) feasible patterns is generally only a subset of the space determined by (1.17), so that fulfillment of conditions in (1.19) is usually just necessary but not sufficient for the existence of a pattern satisfying the constraints. Anyway, there exist two cases, wherein the criterion is sufficient, *i.e.*, the case of uniform linear arrays, and the case of planar arrays whose mask is factorizable as the product of two masks, one along each principal cut [83]. For planar arrays, even if (1.16) satisfies (1.19), since in general it cannot be factorized it does not represent a physically feasible squared amplitude distribution, so that the existence criterion of only necessary. However, in the aforementioned case of factorizable masks, one can use for each principal cut the procedure for the linear case [83], therefore making (1.19) sufficient. Note that in this case the criterion is sufficient but not necessary, because it looks for factorizable excitations, which are just a subset of all the possible ones. In the general case, wherein sufficiency is not guaranteed, the criterion can be used to discard those problems which are certainly unfeasible. In the feasible cases, the pattern furnished by the criterion will be quite certainly not synthesizable. However, exploitation of representation (1.16) allows to state the power synthesis problem in a linear space as small as possible, thus drastically squeezing the set of patterns one should look for with respect to the much larger set of all generic functions compatible with the constraints.

2.4 Planar array tapering

One of the major advantages of array antennas is that the array excitations can be closely controlled to produce extremely low SideLobe Level (SLL) patterns or very accurate approximations of chosen radiation patterns. With reference to radar applications, since in Transmission (T_x) the TransReceiver Modules (TRMs) are usually set to maximum power, we intrinsically associate the tapering of antenna, *i.e.*, the synthesis of its weights, to Receiving (R_x) configurations.

A general synthesis procedure is due to Schelkunov [89] and makes use of the polynomial form of the array factor in the complex variables z_u and z_v:

$$\begin{aligned} z_u &= e^{jk_0(u-u_0)d_x} \\ z_v &= e^{jk_0(v-v_0)d_y} \end{aligned} \tag{1.20},$$

so that (1.11) becomes:

$$AF(\vartheta,\phi) = AF(z_u, z_v) = \sum_{n=1}^{N}\sum_{m=1}^{M} w_{nm} z_u^n z_v^m \tag{1.21}.$$

2.4.1 An example of array synthesis: Dolph-Chebyshev tapering

The procedure commonly referred to as Dolph-Chebyshev synthesis [88] equates the array polynomials to a Chebyshev polynomial and produces the narrowest BeamWidth (BW) subject to a given (constant) side-lobe level, or the lowest SLL for a given BW. Hence it is usually referred to as an optimal array. By assuming factorizable weights (one set for the rows and one for the columns) (1.21) becomes:

$$AF(z_u, z_v) = \sum_{n=1}^{N} w_n z_u^n \sum_{m=1}^{M} w_m z_v^m = AF_x(z_u)\cdot AF_y(z_v) \tag{1.22}.$$

In the hypothesis that $\frac{\lambda}{2} \le d_x \le \lambda$ we obtain [82]:

$$\begin{cases} AF_x(z_u) = \begin{cases} \cos\left(\mathrm{N}\cos^{-1} z_u\right) & for\ |z_u| \le 1 \\ \cosh\left(\mathrm{N}\cosh^{-1} z_u\right) & for\ |z_u| \ge 1 \end{cases} \\ AF_y(z_v) = \begin{cases} \cos\left(\mathrm{M}\cos^{-1} z_v\right) & for\ |z_v| \le 1 \\ \cosh\left(\mathrm{M}\cosh^{-1} z_v\right) & for\ |z_v| \ge 1 \end{cases} \end{cases} \tag{1.23},$$

where

$$\begin{aligned} z_u &= z_1 \cos\left(\frac{\pi d_x \sin\vartheta}{\lambda}\right) \\ z_1 &= \cosh\left(1/\,\mathrm{N}\cosh^{-1} r_1\right) \\ z_v &= z_2 \cos\left(\frac{\pi d_y \sin\vartheta}{\lambda}\right) \\ z_2 &= \cosh\left(1/\,\mathrm{M}\cosh^{-1} r_2\right) \end{aligned} \tag{1.24},$$

and $r_1, r_2 > 1$ such that $20log_{10}r_1$ and $20log_{10}r_2$, which represent the desired SLLs in dB, are positive numbers. For instance, Figure 3, Figure 4, Figure 5, and Figure 6 show, respectively, the normalized 2D power pattern, the normalized tapering, the normalized horizontal and vertical cuts along the *u* and *v* axis of a 15x15 square array (of size 7λ) with distances of half-wavelength and r_1=30dB, r2=40dB.

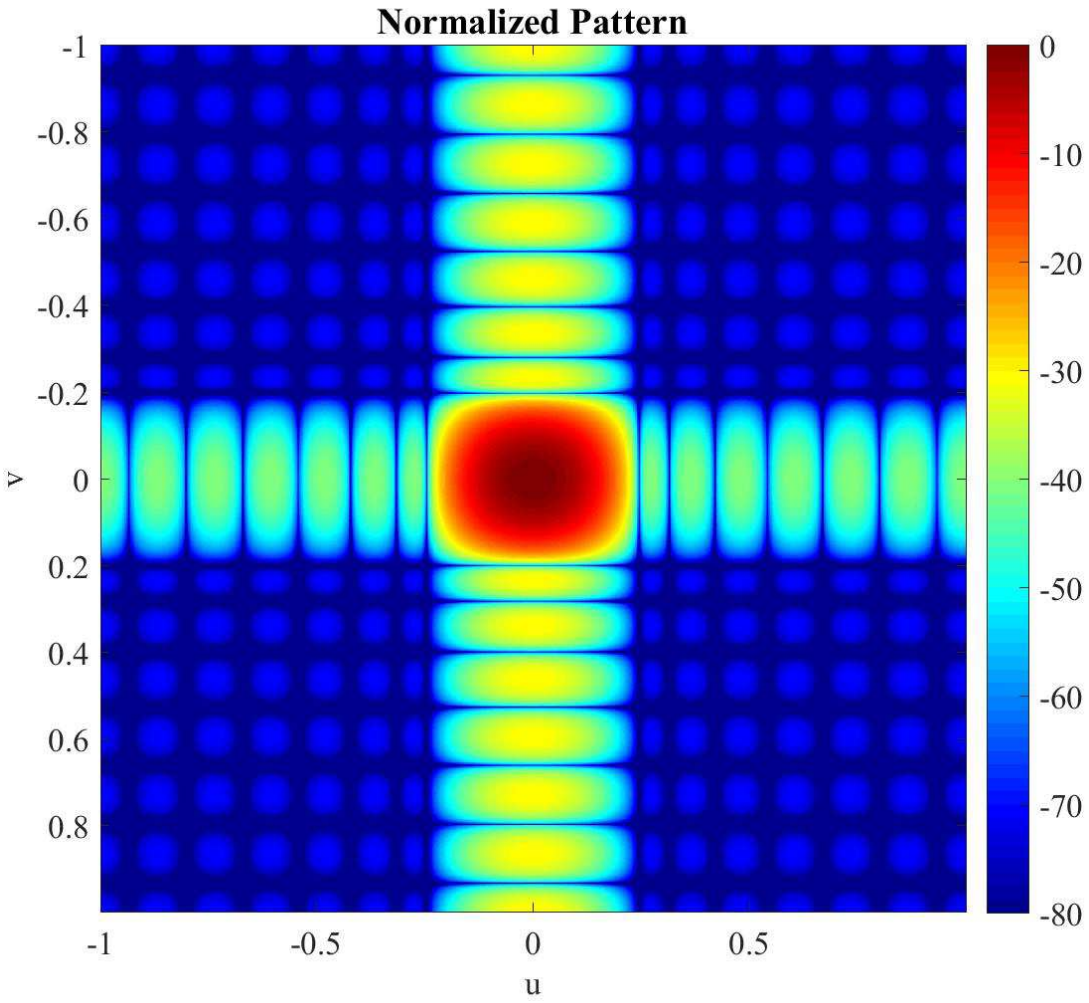

Figure 3. Chebyshev normalized 2D power pattern.

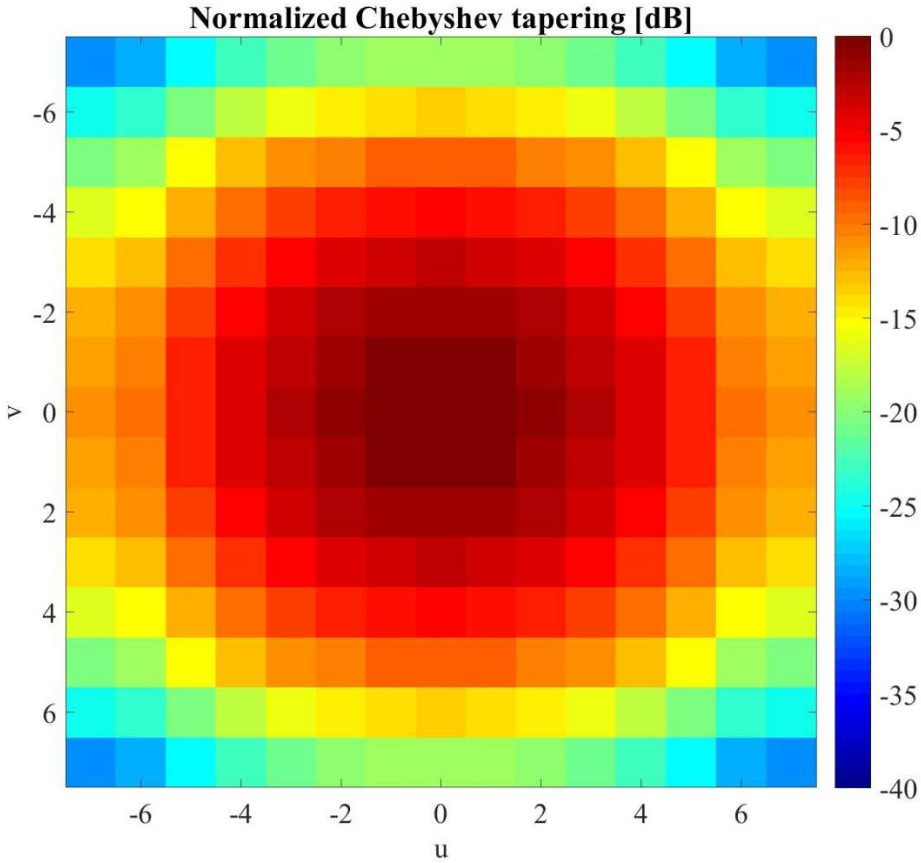

Figure 4. Normalized Chebyshev tapering.

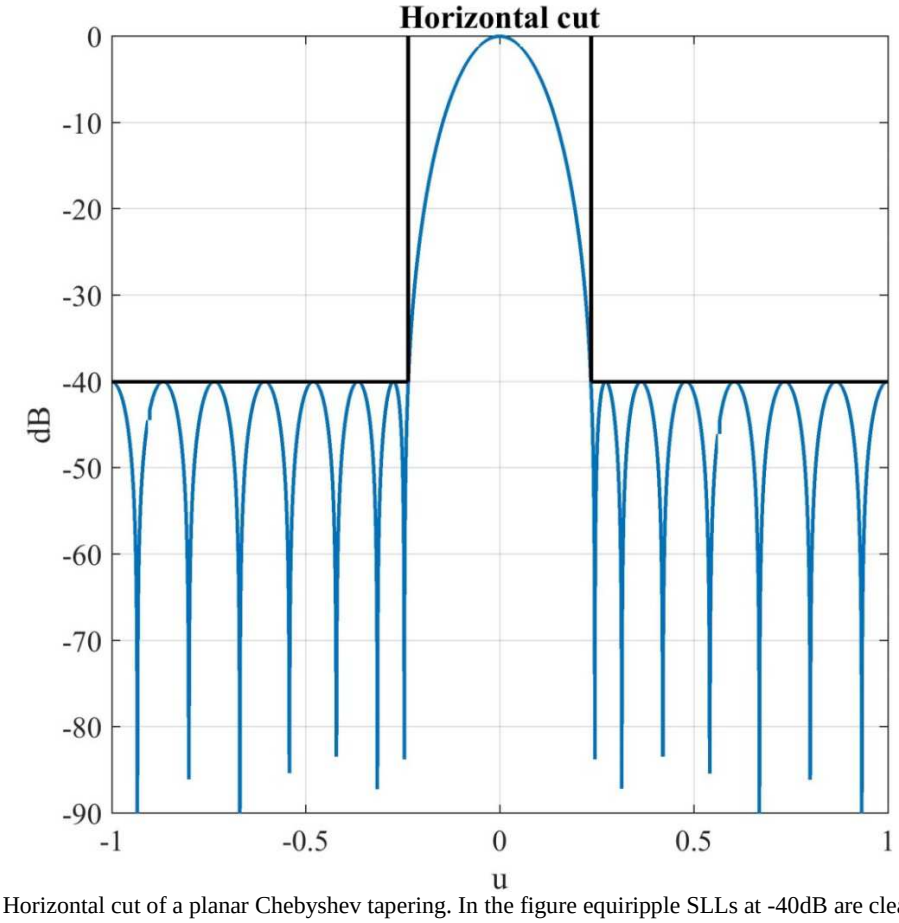

Figure 5. Horizontal cut of a planar Chebyshev tapering. In the figure equiripple SLLs at -40dB are clearly visible

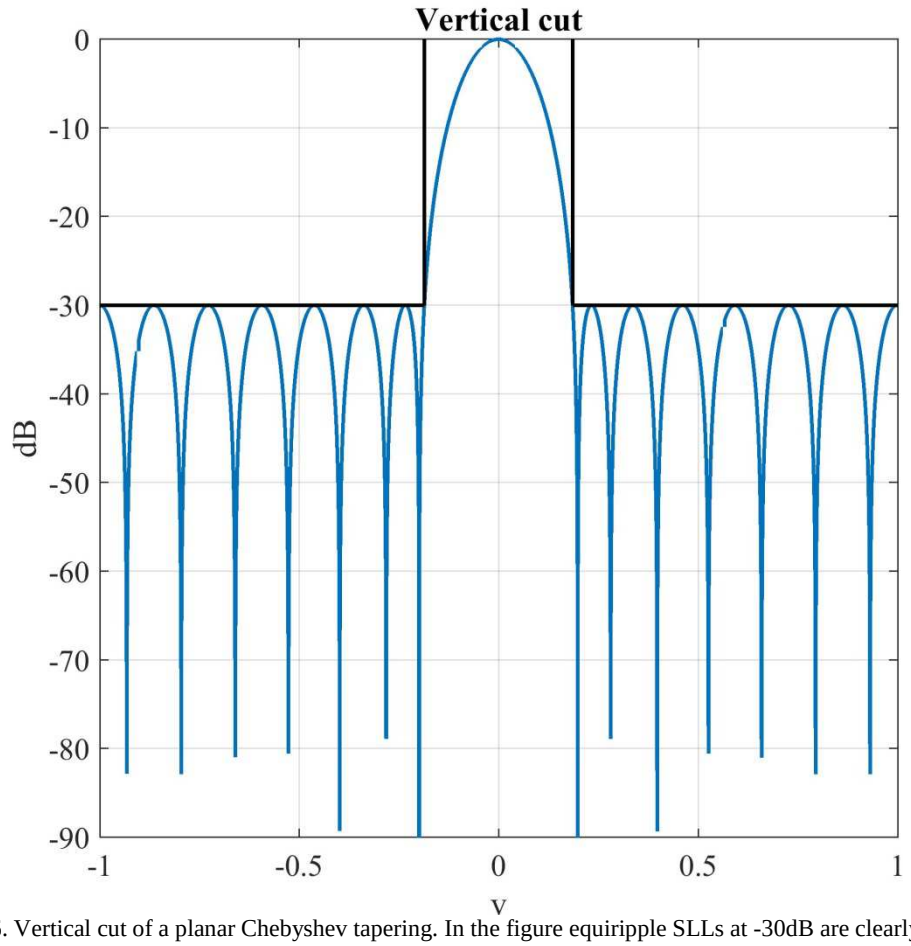

Figure 6. Vertical cut of a planar Chebyshev tapering. In the figure equiripple SLLs at -30dB are clearly visible

In Figure 5 and Figure 6 the SLLs at, respectively, -40dB and -30dB can be easily seen to be at the same level (equiripple).

2.4.2 An example of array synthesis: Taylor tapering

In [90] Taylor analyzed the deficiencies of the Chebyshev pattern and formulated a pattern function that has good efficiency for large arrays. He examined the limit of a continuous line source and drew conclusions about allowed illuminations and pattern SLLs. In particular, he compared the pattern of the 1D Chebyshev illumination with that of a constant illumination *sinc(πz)*, where $z=Nd_x/\lambda$, which has the highest efficiency in the large-array limit. As pointed out by Taylor, the loss in efficiency of the Chebyshev pattern results from the requirement that sidelobe heights are constant. For large arrays, this implies that increasingly more of the energy is in the sidelobe region. In the limit of a very large array, maintaining the Chebyshev sidelobe structure requires an unrealizable aperture illumination. He showed that the far sidelobes of a given line source are a function only of the line source edge illumination. In particular, for a line source of length *2a*, and if the edge illumination has the behavior

$$\left(a-|x|\right)^{\alpha} \tag{1.25}$$

for *x* measured from the center of the source, then for $\alpha \geq 0$, the SLL has the behavior indicated in Table 1. The values for $\alpha < 0$ are not given because the illuminations are unrealizable [82]. The above data show that selecting an aperture illumination with α=0 leads to sidelobes with angular dependence of the type *sinc(πz)*, like those of the uniform illumination. This pattern distribution maintains its efficiency as the array is made larger. Choice of larger values of α makes the sidelobes decay faster, as indicated in Table 1, but have generally lower efficiency.

α	*Asymptotic F(z)*
0	$sinc(\pi z)$
1	$cos(\pi z)/(\pi z)^2$
2	$sin(\pi z)/(\pi z)^3$
3	$cos(\pi z)/(\pi z)^4$

Table 1. Array SLL *Versus* edge illumination parameter α

Taylor also showed that the location of the zeros of the pattern are determined by the edge illumination. The n^{th} pair of pattern zeros (for n large, thus away from the main lobe) occurs at locations $z_n = \pm\left(n + \frac{\alpha}{2}\right)$ for $n \to \infty$. Clearly, this is also consistent with the uniform illumination case for $\alpha=0$. However, when compared to the actual location of the n^{th} pair of zeros for the Chebyshev pattern, it is found that these occur asymptotically at $\pm\left(n - \frac{1}{2}\right)$. Such zero locations correspond to α=-1, an unrealizable illumination for the continuous aperture case.

Taylor expanded upon these mathematical insights to suggest a pattern function with zeros far from the main beam at locations that correspond to the uniform illumination, while the zeros closer to the main beam are chosen similar to those of the Chebyshev pattern. Since Taylor chose to simulate and then modify not the Chebyshev array pattern, but that of a continuous source with features similar to the latter, he used the following ideal line source as substitute:

$$F_0(z,A) = \begin{cases} \cos\left[\pi\sqrt{\left(z^2 - A^2\right)}\right] & \text{for } z^2 > A^2 \\ \cosh\left[\pi\sqrt{\left(A^2 - z^2\right)}\right] & \text{for } z^2 < A^2 \end{cases} \tag{1.26}$$

where, again, $z=Nd_x/\lambda$ and the sidelobe ratio is the value of F_0 at $z=0$, or $r=\cosh(\pi A)$, so that A is defined as

$$A = \frac{1}{\pi}\cosh^{-1} r \tag{1.27}.$$

As shown by Van der Mass [91] this pattern corresponds to the limiting case of the Chebyshev array as the number of elements is indefinitely increased, and has zeros at the locations

$$z_N = \pm\sqrt{\left[A^2 + \left(N - \frac{1}{2}\right)^2\right]}, \quad \mathrm{N} = 1,2,3,\ldots,\infty \tag{1.28}.$$

The pattern has the Chebyshev characteristics with all equal sidelobes, but is physically unrealizable for the reason described earlier, since the far nulls have asymptotic locations corresponding to α=-1. However, although the idealized pattern is unrealizable, Taylor recognized that by selecting a new function with near zeros very close to those of the ideal pattern (1.28) but with zeros corresponding to those of the $sinc(\pi z)$ function at integer values of z, he could satisfy the requirement on both near and far sidelobes. Taylor chose to keep all nulls at the integer location for $|u| \geq \bar{n}$, and to move those for $|u| < \bar{n}$ near the locations (1.28) that would produce the nearly constant sidelobes near the main beam. To match these two sets of zeros, Taylor introduced a dilation factor σ that is slightly greater than unity to stretch the ideal space factor horizontally by moving the ideal zero locations z_n, such that eventually one of the zeros becomes equal to the corresponding integer $\bar{n}$.

The synthesized pattern normalized to unity is

$$F\left(z, A, \bar{n}\right) = \frac{\sin \pi z}{\pi z} \prod_{n=1}^{\bar{n}-1} \frac{1 - z^2 / z_n^2}{1 - z^2 / n^2} \tag{1.29}$$

for $z=Nd_x/\lambda$. The numbers z_n are the zero locations of the synthesized pattern and are given by

$$z_n = \begin{cases} \pm\sigma\sqrt{\left[A^2 + \left(n - \frac{1}{2}\right)^2\right]} & for\, 1 \le n \le \bar{n} \\ \pm n & for\, \bar{n} \le n \le \infty \end{cases} \tag{1.30}$$

where

$$\sigma = \frac{\bar{n}}{\sqrt{\left[A^2 + \left(\bar{n} - \frac{1}{2}\right)^2\right]}} \tag{1.31}.$$

Note that $n = \bar{n}$ at $z_n = \bar{n}$. The aperture distribution to produce Taylor patterns is expanded as a finite Fourier series of terms with zero at the aperture edges [82]:

$$g(x) = F(0, A, \bar{n}) + 2\sum_{m=1}^{\bar{n}-1} F(m, \mathrm{A}, \bar{n}) \cos\left(\frac{2m\pi x}{Nd_x}\right) \tag{1.32}$$

for $-Nd_x / 2 \le x \le Nd_x / 2$. The coefficients are evaluated to be

$$F(m, \mathrm{A}, \bar{n}) = \frac{\left[(\bar{n}-1)!\right]^2}{(\bar{n}-1+m)!(\bar{n}-1-m)!} \prod_{n=1}^{\bar{n}-1}\left[1 - \frac{m^2}{z_n^2}\right] \tag{1.33}.$$

It is important that the distribution be sampled at points one-half spacing from the end of the Taylor distribution function, so the aperture illumination is sampled at the points $(d_x/\lambda)i$ for $\pm i = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \cdots, \frac{(N-1)}{2}$ for arrays with an even number of elements, and $\pm i = 0, 1, 2, \ldots, (N-1)/2$ for arrays with an odd number of elements. The choice of the parameter $\bar{n}$ is not arbitrary, since increasing it retains more of the sidelobes at the design SLL and thus makes the Taylor pattern closer to the Chebyshev one. Increasing $\bar{n}$ thus leads to narrower main beam patterns and higher aperture efficiency, but eventually to aperture illuminations that are not monotonic and have increased illumination near the aperture edges.
By assuming, again, factorizable sets of weights (→ (1.22)) we have evaluated the Taylor distribution for a 29x29 square array (14λ) with distances of half-wavelength with $\bar{n} = 3$ and $\bar{n} = 5$, r_1=30dB, r2=40dB, respectively for the vertical and horizontal cuts.
Figure 7, Figure 8, Figure 9 and Figure 10 show, respectively, the normalized 2D power pattern, the normalized tapering and the normalized horizontal and vertical cuts along the *u* and *v* axis.

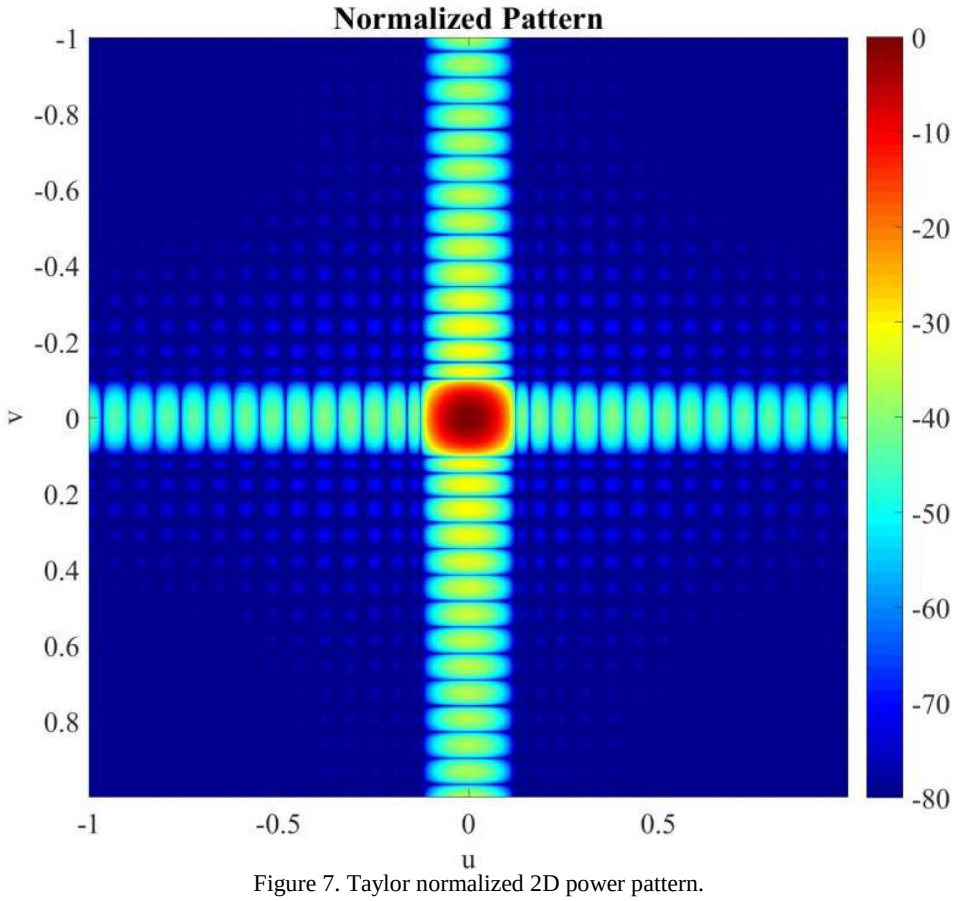

Figure 7. Taylor normalized 2D power pattern.

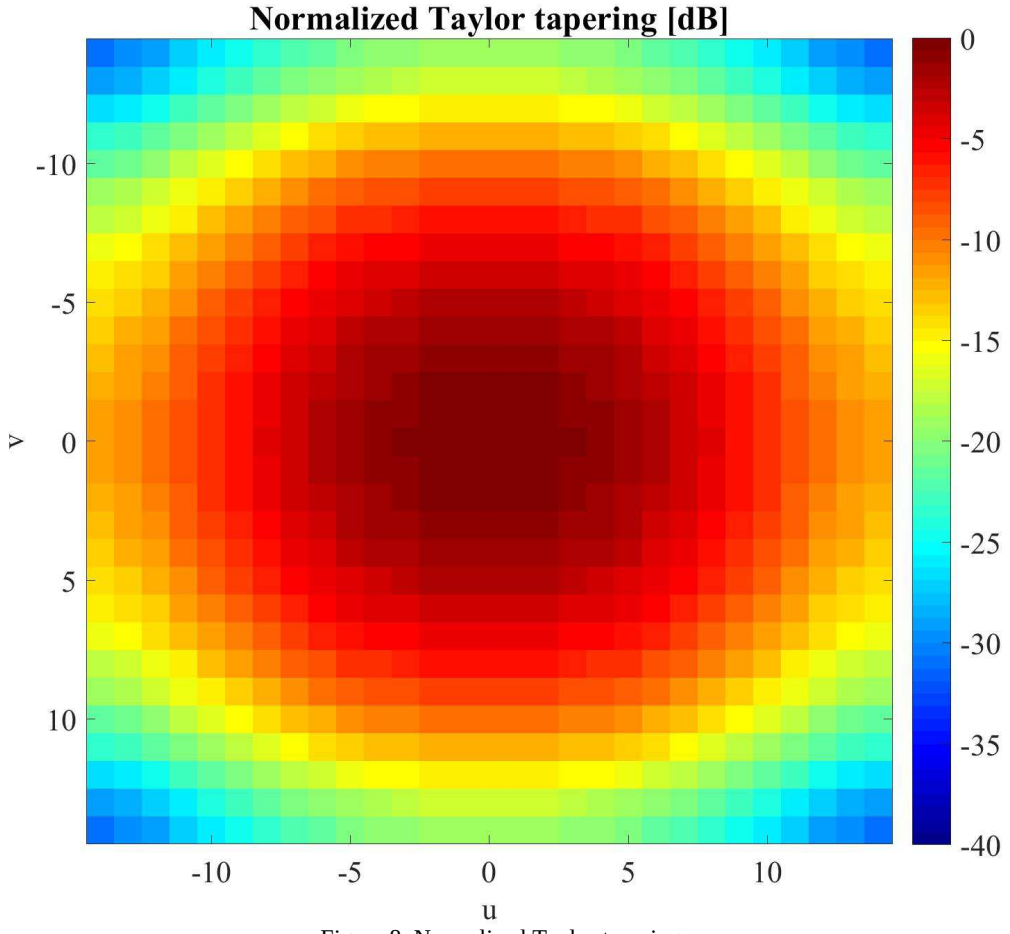

Figure 8. Normalized Taylor tapering.

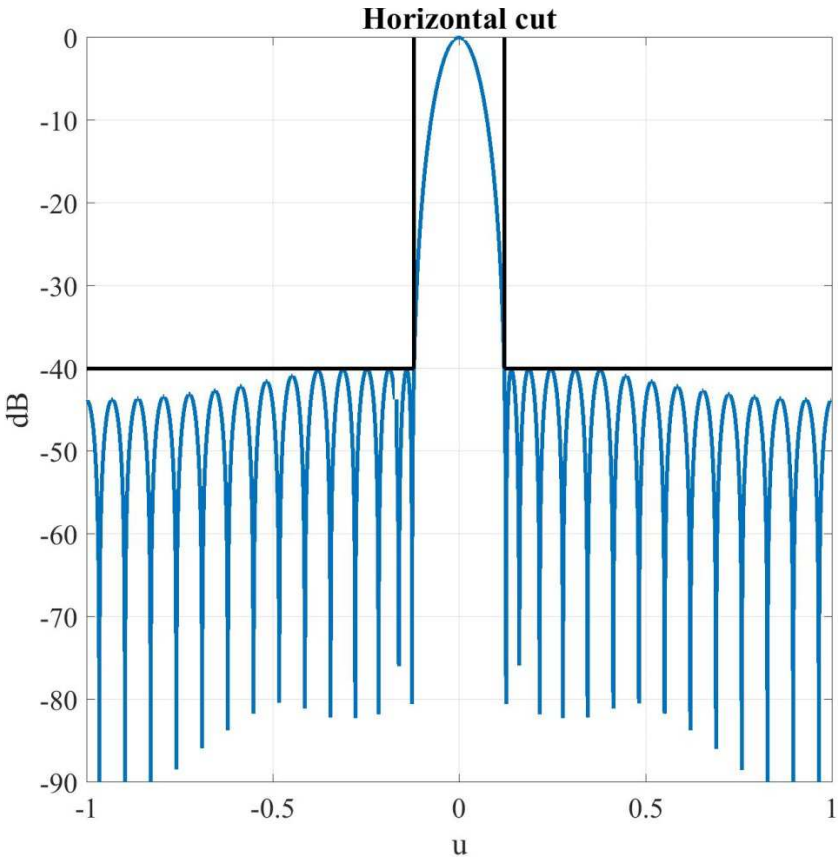

Figure 9. Horizontal cut of a planar Taylor tapering. In the figure it can be seen that the first 5 sidelobes are Chebyshev-like ($\bar{n}=5$), whereas from the sixth one a sinc-type decay begins

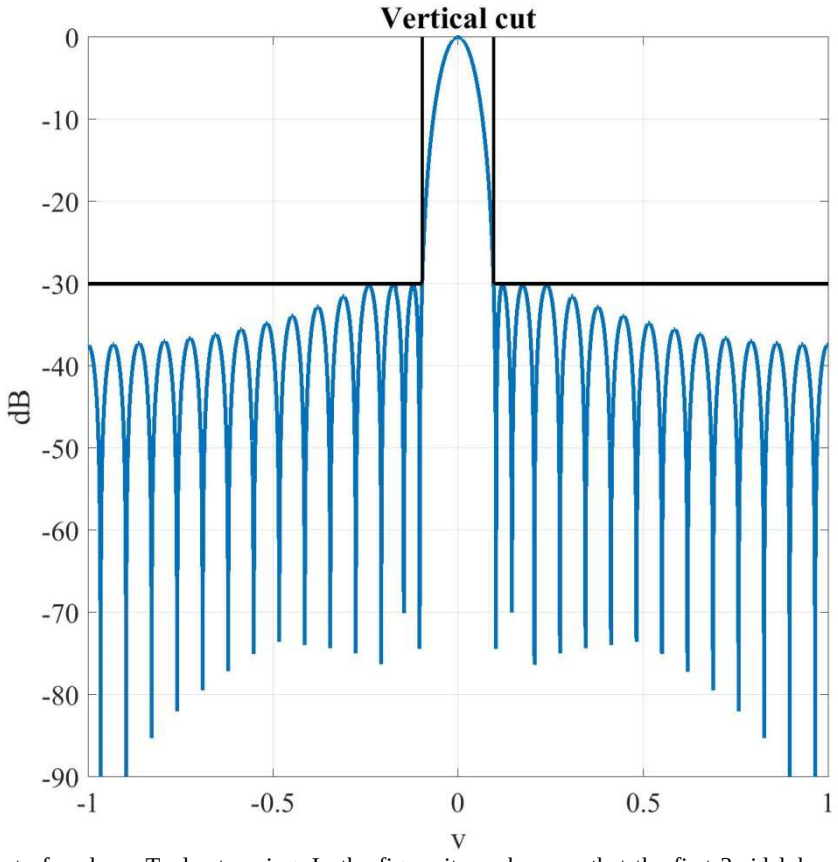

Figure 10. Vertical cut of a planar Taylor tapering. In the figure it can be seen that the first 3 sidelobes are Chebyshev-like ($\bar{n}=3$), whereas from the fourth one a sinc-type decay begins

In Figure 9 and Figure 10 the SLLs at, respectively, -40dB and -30dB can be easily seen to be at the same level (equiripple) for the first 5 and 3 lobes respectively, whereas from the sixth and fourth one a sinc-type decay begins.

2.5 Convex Optimization

In this paragraph we are giving some definitions in the convex optimization framework that will be utilized in the next chapters. In fact we will be talking, apart from convex optimization, about affine conditions, convex hulls, etc, so that the following notions have become necessary.

Given a set $\mathbf{x}=(x_1,..,x_n)\in\mathbf{R}^n$ of variables, a function $f_0:\mathbf{R}^n\to\mathbb{R}$, a set of functions $f_i:\mathbf{R}^n\to\mathbb{R}$ and constants $b_i,\dots,b_m$ for *i=1,…,m*, a convex optimization problem is one of the form:

$$\begin{aligned} &Minimize\left[f_0(\mathbf{x})\right] \\ &subject\ to\ (s.t.) \\ &f_i(\mathbf{x})\le b_i,\ for\ i=1,\dots,m \end{aligned} \tag{1.34},$$

where $f_0(\mathbf{x})$ is the cost function and $f_i(\mathbf{x})$ are the constraints (functions), which are all convex, *i.e.*

$$f_i(\alpha\mathbf{x}+\beta\mathbf{y})\le\alpha f_i+\beta f_i(\mathbf{y}) \tag{1.35}$$

for all $\mathbf{x},\mathbf{y},\in\mathbf{R}^n$ and $\alpha,\beta\in\mathbb{R}$ with $\alpha\ge 0,\ \beta\ge 0,\ \alpha+\beta=1$. $b_i,\dots,b_m$ are the bounds for the constraints. In the following we will consider the contracted notation "*min*" in place of "*minimize*".

2.5.1 Lines and segments

Suppose $\mathbf{x}_1\ne\mathbf{x}_2$ are two points in $\mathbf{R}^n$. Points of the form

$$\mathbf{y}=\vartheta\mathbf{x}_1+(1-\vartheta)\mathbf{x}_2 \tag{1.36},$$

where $\vartheta\in\mathbb{R}$ and $0\le\vartheta\le 1$, form the line passing through $\mathbf{x}_1$ and $\mathbf{x}_2$. Values of the parameter ϑ between 0 and 1 correspond to the (closed) line segment between $\mathbf{x}_1$ and $\mathbf{x}_2$.
Expressing **y** in the form

$$\mathbf{y}=\mathbf{x}_2+\vartheta(\mathbf{x}_1-\mathbf{x}_2) \tag{1.37}$$

gives another interpretation [55]: **y** is the sum of the base point $\mathbf{x}_2$ (corresponding to $\vartheta=0$) and the direction $\mathbf{x}_1$-$\mathbf{x}_2$ (which points from $\mathbf{x}_2$ to $\mathbf{x}_1$) scaled by the parameter ϑ.

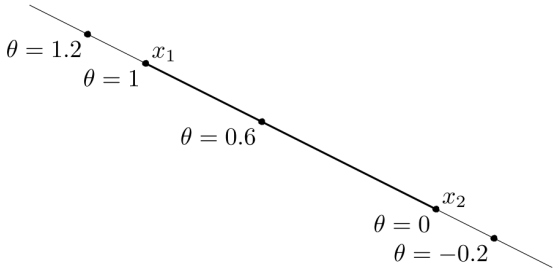

Figure 11. The line passing through $\mathbf{x}_1$ and $\mathbf{x}_2$ is described parametrically by $\vartheta\mathbf{x}_1+(1-\vartheta)\mathbf{x}_2$, where ϑ varies over $\mathbb{R}$. The line segment between $\mathbf{x}_1$ and $\mathbf{x}_2$, which corresponds to ϑ between 0 and 1, is shown darker.

Thus ϑ gives the fraction of the way from $\mathbf{x}_2$ to $\mathbf{x}_1$ where **y** lies. As ϑ increases from 0 to 1 the point **y** moves from $\mathbf{x}_2$ to $\mathbf{x}_1$; for $\vartheta>1$, the point **y** lies on the line beyond $\mathbf{x}_1$ (Figure 11).

2.5.2 Affine sets

A set $C \subseteq \mathbf{R}^n$ is affine if the line through any two distinct points in *C* lies in *C*, *i.e.* if for any $\mathbf{x}_1, \mathbf{x}_2 \in C$ and $\vartheta \in \mathbb{R}$, we have $\vartheta\mathbf{x}_1 + (1-\vartheta)\mathbf{x}_2 \in C$. Actually *C* contains the linear combination of any two points in *C*, provided the coefficients in the linear combination sum to one. The idea can be generalized to more than two points: a point of the form $\vartheta_1\mathbf{x}_1 + \ldots + \vartheta_k\mathbf{x}_k$, where $\vartheta_1 + \ldots + \vartheta_k = 1$, is an affine combination of the k points $\mathbf{x}_1,\ldots,\mathbf{x}_k$. It can be shown [55] that an affine set contains every affine combination of its points.

2.5.3 Convex sets

A set $C \subseteq \mathbf{R}^n$ is convex if the line segment between any two points in *C* lies in *C*, *i.e.* if for any $\mathbf{x}_1, \mathbf{x}_2 \in C$ and any ϑ such that $0 \le \vartheta \le 1$, we have $\vartheta\mathbf{x}_1 + (1-\vartheta)\mathbf{x}_2 \in C$. In practice a set is convex if every point in the set can be seen by every other point along an unobstructed straight path between them, where "unobstructed" means lying in the set. Clearly, every affine set is also convex since it contains the entire line between any two distinct points in it, and therefore also the line segment between the points.
A convex combination of the points $\mathbf{x}_1,\ldots,\mathbf{x}_k$ is a point of the form $\vartheta_1\mathbf{x}_1 + \ldots + \vartheta_k\mathbf{x}_k$ where $\vartheta_1 + \ldots + \vartheta_k = 1$ and $\vartheta_i \ge 0$ for *i=1,…,k*. It can be shown [55] that a set is convex if and only if it contains every convex combination of its point.

2.5.4 The convex hull

The convex hull of a set $C \subseteq \mathbf{R}^n$, denoted "*conv(C)*" is the set of all convex combinations of points in C:

$$conv(C) = \left\{ \vartheta_1\mathbf{x}_1 + \ldots + \vartheta_k\mathbf{x}_k \mid \mathbf{x}_i \in C, \vartheta_i \ge 0, i = 1,\ldots,k, \sum_{i=1}^{k} \vartheta_i = 1 \right\} \qquad (1.38).$$

As the name suggests, such set is always convex and in particular it is the smallest convex set that contains *C*.

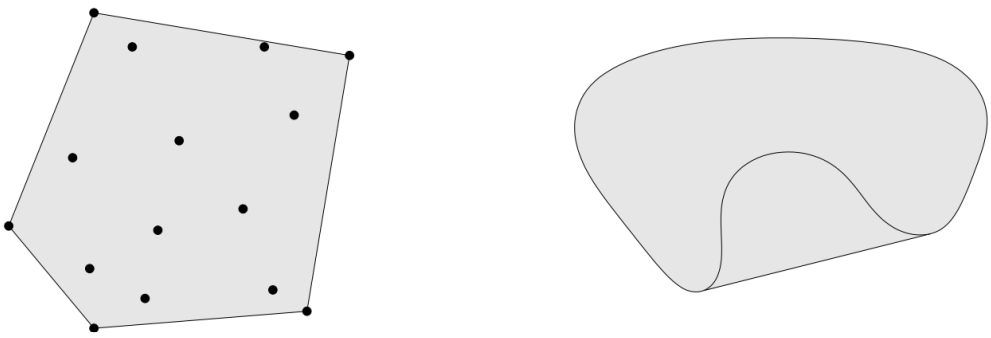

Figure 12. The convex hulls of two sets in $\mathbb{R}^2$. *Left.* The convex hull of a set of fifteen points (shown as dots) is the pentagon that encompasses them. *Right.* The convex hull of a non convex set.

Figure 12 shows an example of convex hulls of a convex (left) and non convex (right) set.

2.5.5 Cones

A set $C \subseteq \mathbf{R}^n$ is called a cone if for every $\mathbf{x} \in C$ and $\vartheta \geq 0$ we have $\vartheta \mathbf{x} \in C$. A set *C* is a convex cone if it is convex and a cone, which means that for any and $\mathbf{x}_1, \mathbf{x}_2 \in C$ and $\vartheta_1, \vartheta_2 \geq 0$, we have $\vartheta_1 \mathbf{x}_1 + \vartheta_2 \mathbf{x}_2 \in C$. Points of this type can be described geometrically as forming the two-dimensional pie slice with apex 0 and edges passing through $\mathbf{x}_1$ and $\mathbf{x}_2$ (see Figure 13).
A conic combination (or a nonnegative linear combination) of the points $\mathbf{x}_1, \ldots, \mathbf{x}_k$ is a point of the form $\vartheta_1 \mathbf{x}_1 + \ldots + \vartheta_k \mathbf{x}_k$ where $\vartheta_1 + \ldots + \vartheta_k = 1$ and $\vartheta_i \geq 0$ for *i=1,…,k*. It can be shown [55] that a set is a convex cone if and only if it contains all conic combinations of its elements.

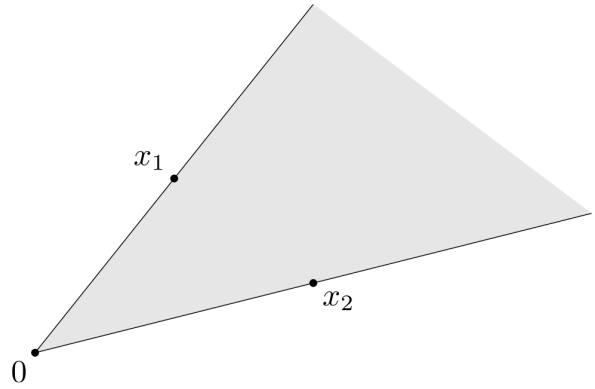

Figure 13. The pie slice shows all points of the form $\vartheta_1 \mathbf{x}_1 + \vartheta_2 \mathbf{x}_2$, where $\vartheta_1, \vartheta_2 \geq 0$. The apex of the slice (which corresponds to $\vartheta_1 = \vartheta_2 = 0$) is at 0; its edges (which correspond to $\vartheta_1 = 0$ or $\vartheta_2 = 0$) pass through the points $\mathbf{x}_1$ and $\mathbf{x}_2$.

2.5.6 Norms and quasi-norms

A function $f : \mathbf{x} \in \mathbf{R}^n \rightarrow \|\mathbf{x}\| \in \mathbb{R}$ with *dom(f)*=$\mathbf{R}^n$ is called a norm if

- *f* is nonnegative: ***f(x)≥0*** for all $\mathbf{x} \in \mathbf{R}^n$.
- *f* is definite: ***f(x)=0*** only if **x**=**0**.
- *f* is homogeneous: *f(t**x**)=|t|f(**x**)*, for all $\mathbf{x} \in \mathbf{R}^n$ and $t \in \mathbb{R}$.
- *f* satisfies the triangle inequality: $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$.

An ℓ_p norm on the vector **x** is defined by:

$$\|\mathbf{x}\|_p = \left(|x_1|^p + \ldots + |x_n|^p \right)^{1/p} \tag{1.39}$$

for $p \geq 1$.

For $0<p<1$ the (1.39) is only a quasi-norm, meaning that *f* meets the same properties of the norms except the triangular inequality, which is substituted by the relation:

$$\|\mathbf{x}+\mathbf{y}\| \leq K\left(\|\mathbf{x}\| + \|\mathbf{y}\|\right) \tag{1.40}$$

for some constant K≥1.

2.5.7 Norm balls and norm cones

From the general properties of norms it can be shown that a norm ball of radius r and center $\mathbf{x}_c$, given by $\{\mathbf{x} \mid \|\mathbf{x}-\mathbf{x}_c\| \le r\}$, is convex [55]. The norm cone associated with the norm $\|\cdot\|$ is the set

$$D_{\|\cdot\|} = \{(\mathbf{x},t) \mid \|\mathbf{x}\| \le t\} \subseteq \mathbf{R}^{n+1} \tag{1.41}.$$

As the name suggests it is a convex cone. An important case of norm cone is the second-order cone, *i.e.* the norm cone for the Euclidean norm:

$$D_2 = \{(\mathbf{x},t) \in \mathbf{R}^{n+1} \mid \|\mathbf{x}\|_2 \le t\} \tag{1.42}.$$

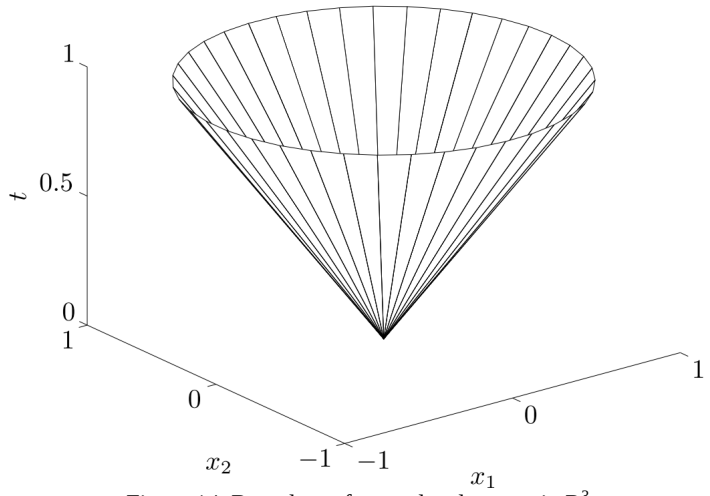

Figure 14. Boundary of second-order cone in R^3

Figure 14 depicts the second-order cone in $\mathbf{R}^3$: $\{(x_1, x_2, t) \mid \sqrt{x_1^2 + x_2^2} \le t\}$.

The set of all vectors with norm less than or equal to one,

$$B = \{\mathbf{x} \in \mathbf{R}^n \mid \|\mathbf{x}\| \le 1\} \tag{1.43},$$

is called the unit ball of the norm $\|\cdot\|$. The unit ball satisfies the following properties:

- B is symmetric about the origin, i.e. $\mathbf{x} \in B$ if and only if $-\mathbf{x} \in B$.
- B is convex.
- B is closed, bounded, and has nonempty interior [55].

Chapter 3

The developed algorithms

In this chapter we describe in detail the developed algorithms based on convex optimization. In particular, the innovative content, their structure, the goals, the problems that come out in their realization, the solutions proposed for the latter and their strengths and weaknesses will be addressed.

3.1 Introduction

The synthesis schemes presented in the following are of mask-constrained type [42] as the idea that we had since the beginning was to apply them for future AESA projects, in which the requirements the FF has to meet are given in terms of masks, regardless of the beam characteristics. In fact, the most important features for a radar like range coverage, PD, PFA, azimuth and elevation BWs, monopulse performances, peak directivity, etc., are all obtained with (or indirectly derived from) lower and upper bounds instead of exact numbers. Moreover, a reference field pattern to match in amplitude and phase would dramatically reduce the number of degrees of freedom the algorithms can pick from. Two approaches will be adopted for the synthesis process: one that falls in the CS framework and that is based on reweighted ℓ_1-norm sequential refinements [73], the other one on MIP. Although several works exist in literature on this kind of synthesis [46-48] [50], very few of them address the synthesis of sparse arrays in terms of minimization of the ℓ_0 pseudo-norm of the number of elements that constitute the initial grid [78] [92]. None of them, to the best of the PHD student's knowledge, takes into account the non-superdirectivity by imposing a constraint on the minimum inter-element distance, which is, indeed, actually one of the most innovative contributes of this work.
The algorithms scheme is simple: we start from an initial array, that we can call the "full array" whose inter-element spacing is half-wavelength and whose shape and size(s) can be any and are fixed a priori (thus the BWs are fixed as well). Then we derive the minimum SLL in a Chebyshev sense, *i.e.* we synthesize a set of weights that achieves the lowest maximum SLL in a bidimensional region that spans from the points on the main lobe that intercept the highest SLL (*i.e.* the main lobe BW at the highest SLL) to the visible boundary. This way we get a main lobe BW at maximum SLL and a SLL which both act as requirements for the sparse synthesis. In order to carry out the latter we sample the full array with wavelength submultiples (for instance $\lambda/10$, or $\lambda/100$), thus obtaining an initial, dense grid of elements, and then minimize the number of elements subject to an ensemble of requirements, among which a requirement on the minimum inter-element distance. The number of elements of this grid are the degrees of freedom the solver has got for the optimization process.
Clearly, the just described scheme works for any type of geometry and the aforementioned "optimum SLL" is only a choice, but in no way it is supposed to be the only one.

3.2 The Optimum SLL

The sparse arrays achieved by optimization by means of our developed algorithms, that will be described later in this chapter, will be compared to a reference array in terms of reduction of the number of elements. Actually, such reference array is the one which the algorithms sparsify on. We can name it as the "full array". Its geometry, lattice and size are fixed a priori and nonetheless can be any. We start from a rectangular geometry and lattice with half-wavelength inter-element distances. The BW is fixed a priori, and the optimum SLL is derived accordingly. By recalling (1.11) the problem is the following:

$$\begin{aligned} &\min_{\mathbf{w}\in\mathbb{R}^{NM}} (SLL) \\ &s.t. \\ &\left|AF(\text{SLL\_region})\right| \le SLL \\ &SLL \in \mathbb{R} \\ &w_{nm} \ge 0 \ \ \forall n = 1,\ldots,N,\ m = 1,\ldots,M \end{aligned} \tag{2.1},$$

where we denoted by **w** the vector containing all the weights w_{nm}, and SLL_region is the bidimensional region of the spectrum external to the intersection between the main lobe (multiplied by the scalar m_c) and the line at the highest SLL. Given this optimum SLL, the size of the aperture that achieves it by tapering is also the smallest non-superdirective source, this way returning a necessary condition for non-superdirectivity [48]. Hence the full array is the minimally sparse solution that meets the SLL requirement. For instance, in Figure 15 and Figure 16, for a 3λ square array with rectangular lattice, the isophoric and tapered horizontal cuts and the illumination are shown respectively for a BW at maximum SLL enlarged by m_c=1.5.

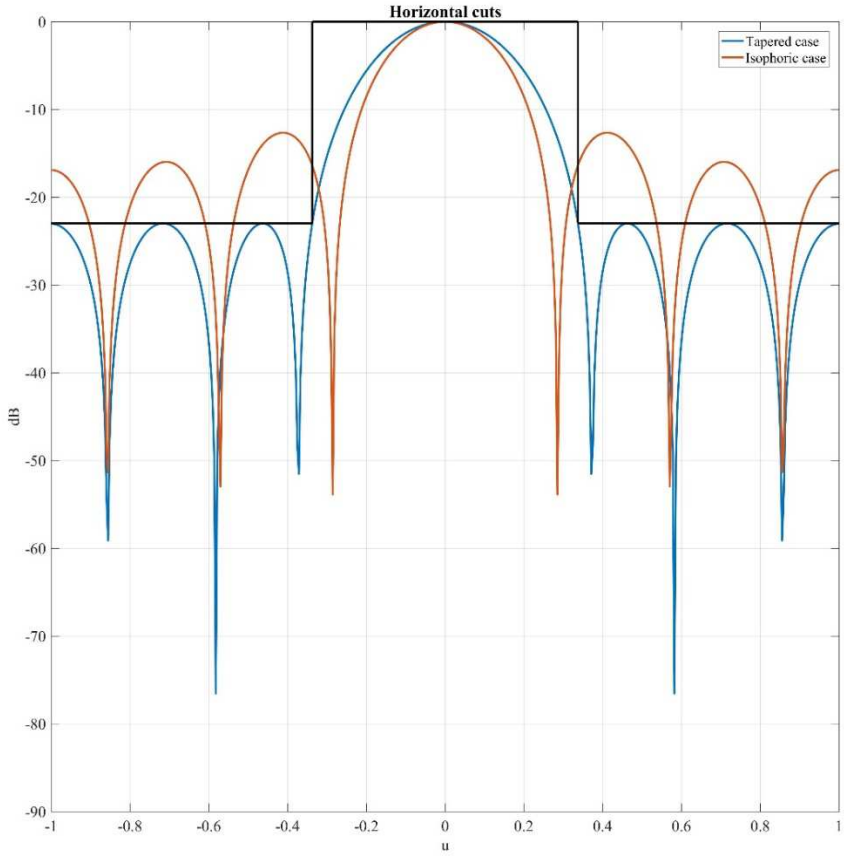

Figure 15. Horizontal cuts of a 3λ square array with rectangular lattice in the tapered (blu line) and isophoric (red) cases. The optimum SLL is -22.9784dB [48].

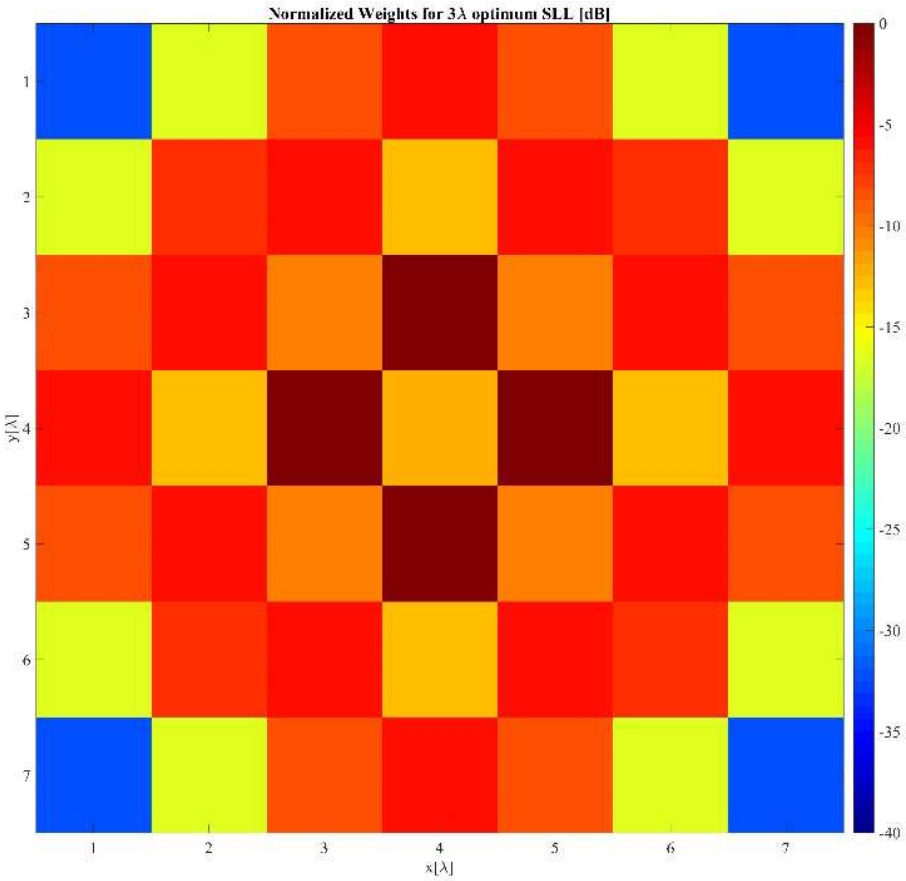

Figure 16. Tapering of a 3λ square array with optimum SLL equal to -22.9784dB [48].

3.3 Problem statement: the minimization of the ℓ_0 pseudo-norm

Once the bounds have been assigned to the desired power pattern, the aim is to find the non-superdirective excitation and location distributions of the radiating elements such that the radiated field satisfies the prescribed power mask with the minimum number of antennas for a desired solid angle of steering. The synthesis problem is formulated as one of constrained optimization type, in which the cost function is chosen to return the solution that exhibits the minimum cardinality, *i.e.* the ℓ_0 pseudo-norm. Such a constraint limits the number of non-zero entries of the array weights, is non convex, and thus requires, in principle, to solve an NP-hard problem [48]. However, by introducing a vector $\mathbf{w}_b$ of binary variables belonging to $B=\{0,1\}^{NM}$ (where the product *NxM* represents the (dense) grid obtained by sampling the full-array at submultiples of λ) accounting for the presence or absence of the actual radiating element, we can easily assure the non super-directivity condition in a convex way, perform a two-way synthesis (either in Tx and Rx) and make the cost function a simple sum of terms, although the very latter, as we are about to see, represents the main difficulty to face. Formally we can express the synthesis problem as:

$$\min_{\mathbf{w}_b \in B} \left(\sum_{i=1}^{NM} w_{b_i} \right)$$

s.t.

$$\begin{aligned}
& LB_k(u,v) \le \left| AF_{Tx}(u,v) \right|_{(u,v)\in\Omega_k} \le UB_k(u,v) \\
& LB_h(u,v) \le \left| AF_{Rx}(u,v) \right|_{(u,v)\in\Phi_h} \le UB_h(u,v) \\
& k = 1,\ldots,N_{T_x} \\
& h = 1,\ldots,N_{R_x} \\
& |\mathbf{w}| \le \mathbf{w}_b \\
& \mathrm{Max}(\mathbf{w}) - 10^{\frac{TR}{20}} \mathrm{Min}(\mathbf{w}) \le 0 \\
& \mathbf{A}\mathbf{w}_b \le \mathbf{1}
\end{aligned} \qquad (2.2),$$

where w_{b_i} is the i[th] entry of $\mathbf{w}_b$, $k=1,\ldots,N_{T_x}$ identifies the k[th] constraint in Tx, N_{T_x} is the number of constraints in Tx, Ω_k is the k[th] region in Tx in the *(u,v)* spectrum (it can be monodimensional or bidimensional, depending on the requirement), $UB_k(u,v)$ is the k[th] upper bound in Tx, $LB_k(u,v)$ is the k[th] lower bound in Tx, $h=1,\ldots,N_{R_x}$ identifies the h[th] constraint in Rx, N_{R_x} is the number of constraints in Rx, Φ_h is the h[th] region in Rx in the *(u,v)* spectrum (it can be monodimensional or bidimensional, depending on the requirement), $UB_h(u,v)$ is the h[th] upper bound in Rx, $LB_h(u,v)$ is the h[th] lower bound in Rx, **A** is the non-superdirectivity matrix (which we are going to discuss in detail soon) of size (*NM*)*x*(*NM*), **w** is the set of all weights in (1.11), **1** is a column vector of *NM* ones and TR, that is a positive quantity, is the maximum dynamics allowed by the TRMs. To each of the relations in (2.2) is being given an exhaustive discussion. Note that some constraints in (2.2) may not be present in each type of synthesis, as well as some of them may be modified for a specific goal (like pencil-type arrays for example). For instance the constraint on the maximum allowed dynamics for the R_x pattern(s) is not required for the comparisons to the literature benchmarks, but it is required for the sparsification of the chosen antenna (see 5.2). Finally, note that the introduction of the binary vector $\mathbf{w}_b$ makes the cost function in problem (2.2) convex and thus exploitable for any type of convex solver, but does not change the fact that it remains NP-hard, therefore such introduction has the only purpose of letting problem (2.2) be practicable for software implementation. In fact we could write the cost function as it is usually found in literature, so without resorting to the vector $\mathbf{w}_b$ and thus involving directly the vector ***w***:

$$\min_{\mathbf{w}} \|\mathbf{w}\|_0 \qquad (2.3).$$

The reason why problem (2.2) is particularly suitable for software implementation lies in how rewriting systems for convex optimization [102] work. Basically, what happens when problem (2.2) is implemented is that a check is carried out for possible convexity violations, then once granted this step the appropriate solver is invoked. If any convexity violation is detected, the solver is not drawn in at all, and the simulation returns an error. Specifically, the cost function in (2.2) is the sum of the entries of the vector $\mathbf{w}_b$, thus it represents no convexity violation if for a moment we do not consider they are integer variables. Only afterwards the rewriting system recognizes the minimization involved in (2.2) recalls a non-convex and NP-hard problem, but at this step the correct solver is requested, and the B&B algorithms come into play. From this point on it is up to the solver to handle the (non convex) mixed integer problem. Hence, provided that there are no convexity violations and a proper solver is available, the final result is that the simulation starts smoothly as well as the solving process.

3.3.1 The steerability

The maximum desired steering in *u* and *v* determines an ellipse in the *(u,v)* spectrum whose major and minor axis extend beyond the visible circle, so that any unwanted lobe is automatically suppressed. For example, in Figure 17 a 7λ square array steerable in azimuth and elevation at ±57° is shown. In particular, the dotted circle represents the farthest boundary to which the upper bound constraints have been extended.

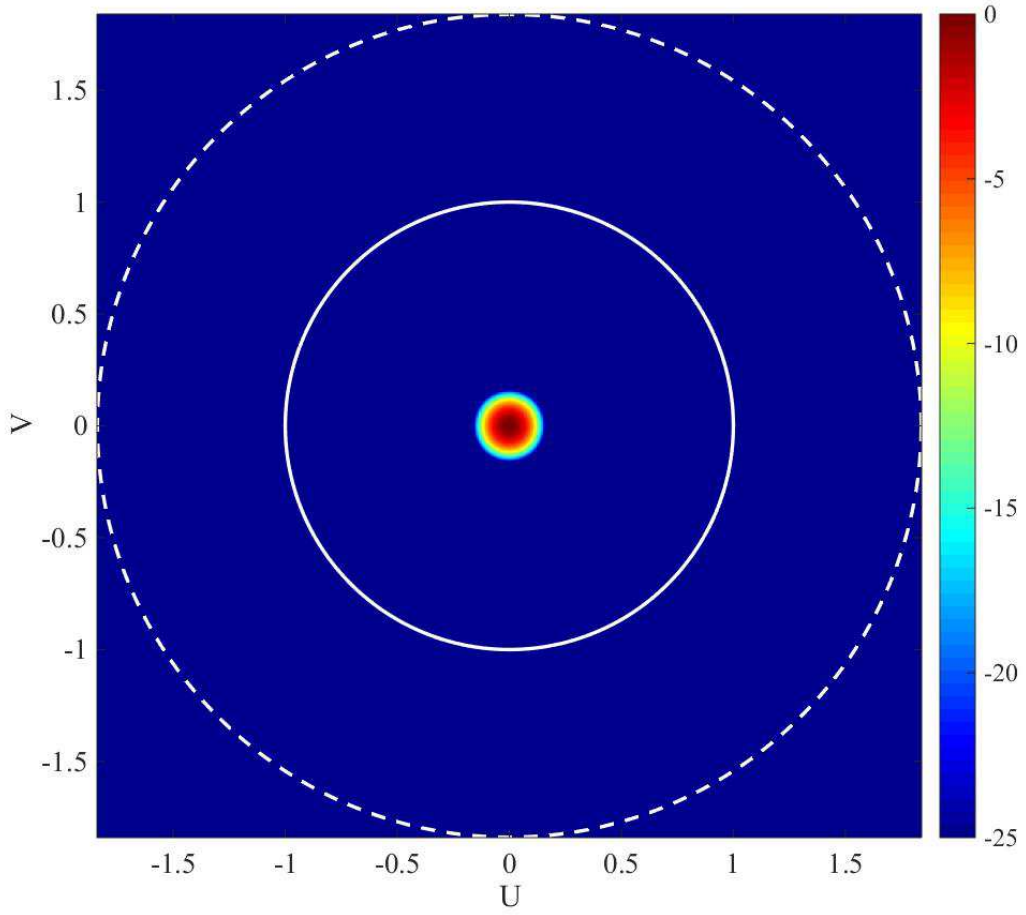

Figure 17. 7λ spectrum (the colourbar is set in such way that only the main lobe is visible). The desired steering is 57° either in azimuth and elevation, so that the ellipse beyond the visible circle is in this case a circle as well of ray 1+sin(57°)≈1.83 (dotted white circle).

3.3.2 Upper and lower bounds

The constraint

$$\left|AF_{R_x}(u,v)\right|_{(u,v)\in\Phi_h} \le UB_h(u,v) \qquad h=1,\ldots,N_{R_x} \tag{2.4}$$

in (2.2) is a special case of Second Order Cone (SOC) constraint whenever the right hand side is a system of linear equations. The general definition of SOC constraint is given by [55]:

$$\left\|\mathbf{Ax}+\mathbf{b}\right\|_2 \le \mathbf{c}^T\mathbf{x}+d \tag{2.5}$$

where $\mathbf{A}\in\mathbf{R}^{kxn}$, $\mathbf{b}\in\mathbf{R}^{k}$, $\mathbf{c}\in\mathbf{R}^{n}$ and $d\in\mathbb{R}$. In particular, the (2.4) corresponds to the (2.5) when c_i=0 for $i=1,\ldots,n$; in such case it can be rewritten as

$$\left\|\mathbf{Ax}+\mathbf{b}\right\|_2^2 \le d^2 \tag{2.6},$$

which is a convex quadratic constraint, efficiently solvable with the interior points method [55].
As regards the left hand inequality of the fourth relation after "s.t." in (2.2), which is typically related to non-pencil pattern types, we observe that in general it is a non-convex constraint [81]. An assumption that can be made to overcome this problem is to adopt conjugate-symmetrical distributions for the pattern that requires to meet the lower bound constraints, so that these become linear inequalities. Hence in this hypothesis the non-convexity of the lower bound can be eliminated. Another valid assumption is to impose the distribution to be real and even. Same considerations apply for Tx (third relation after "s.t." in (2.2)).

3.3.3 Two-ways synthesis

The problem (2.2) intrinsically outputs the same layout for Tx and Rx, with two sets of *NM* weights: in the former case such set is a binary string, whereas in the latter one the weights are real. We call this "two-way synthesis". In fact in transmission mode the power supplied by the single TRM to the radiating element is as high as possible whilst in receiving mode different values of attenuation are set in order to obtain the desired receiving pattern. Often in our company the power mask requirements are given overall, meaning that the SLLs must be below a specified threshold for the upper bounds and above for the lower ones, jointly in Tx and Rx. Generally the SLL upper bounds required in Tx are higher than the ones in Rx (since fewer degrees of freedom are available in this case), thus forcing to synthesize two separate and independent arrays. With our formulation instead it suffices to add the following constraints to (2.2)

$$\begin{aligned} & f_{LB}\left(LB_k(u,v), LB_h(u,v)\right) \ge SLL_{LB} \\ & f_{UB}\left(UB_k(u,v), UB_h(u,v)\right) \le SLL_{UB} \\ & k = 1,\ldots,N_{T_x} \\ & h = 1,\ldots,N_{R_x} \end{aligned} \tag{2.7},$$

where the upper and lower bounds either in Tx and Rx are the independent variables of two functions $f_{LB}\left(LB_k(u,v), LB_h(u,v)\right)$ and $f_{UB}\left(UB_k(u,v), UB_h(u,v)\right)$ which must be greater and smaller than both an overall lower and upper SLL respectively. Clearly, these functions are required to be convex.
The relation $|\mathbf{w}| \le \mathbf{w}_b$ in (2.2) ensures the layout for Tx and Rx patterns is shared. Actually such constraint may be just one of a set of equivalent constraints that account for different Rx patterns which all share the same layout, as is done in [50] for instance.

3.3.4 The non super-directivity matrix

A sparse array is superdirective if its directivity is higher than that of its full version. Excessive array superdirectivity inflicts major problems in low radiation resistance (hence low efficiency), sensitive excitation and position tolerances and narrow bandwidth. Moreover mutual coupling among elements becomes hardly controllable. A condition that ensures non-superdirectivity is that the distance between each couple of elements of the starting grid must be greater than or equal half-wavelength. The non-superdirectivity matrix, that in (2.2) is called "**A**", is defined by means of its general entry [48]:

$$a_{ij} = \begin{cases} 1 \ \forall i, j = 1,\ldots NM : \left\| P_i - P_j \right\|_2 \le \dfrac{\lambda}{4} \\ 0 \ otherwise \end{cases} \tag{2.8},$$

where $P_i = (x_i, y_i)$ and $P_j = (x_j, y_j)$. For each point of the initial grid (see Figure 2) **A** centers a circle of radius λ/4 whose inside is filled with ones, the outside with zeros. Figure 18 and Figure 19 show two rows of **A** (chosen randomly) for a 3λ square array with a sampling of λ/30: the red points are points at distance smaller than half-wavelength with respect to the circle's center. Clearly, such rows have length *NM*, thus they have been reshaped as an *NxM* matrix in order to show the binary images in Figure 18 and Figure 19.

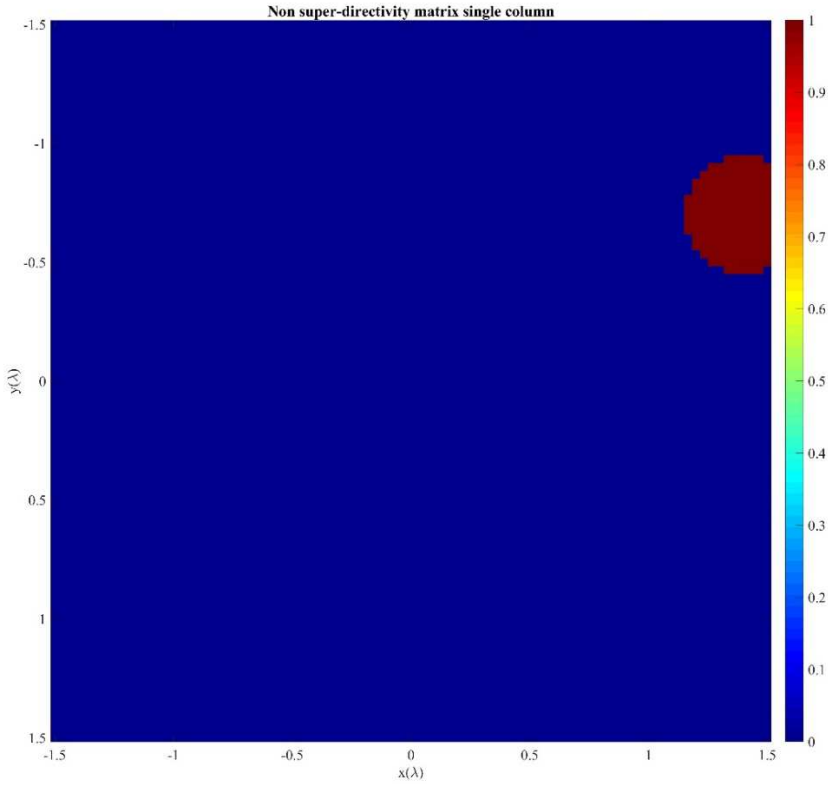

Figure 18. Single column of matrix **A**. The red points represent all the points at distance smaller than half-wavelength with respect to the circle's center, whereas the blue ones represent all the points at distance greater than or equal half-wavelength.

The non-superdirectivity is achieved by constraining each entry of the vector resulting from the product $\mathbf{A}\mathbf{w}_b$, to be smaller than or equal to unity. Formally, if we denote with K_q the set of the grid indexes identifying the elements that fall in the q-*th* circle, the q-*th* entry of the above scalar product is:

$$\mathbf{A}\big|_q * \mathbf{w}_b = r_q = \sum_{i \in K_q} \mathbf{w}_b(i), \quad q \in \{1, 2, \ldots, NM\} \tag{2.9}.$$

The vector resulting from the product $\mathbf{A}\mathbf{w}_b$ exhibits points of the grid with an integer amplitude that is indicative of superdirectivity whenever is greater than 1.

For instance, Figure 20 shows a forced superdirective distribution denoted by $\tilde{\mathbf{w}}_b$ for a square grid of edge 3λ. Four circles related to four specific rows of **A** are highlighted. The first one encompasses no active elements (r_1=0 in this region); the second and third circles represent a boundary case since the two elements they are centered at are exactly λ/2 distant (r_2=r_3=1). The last one is centered at a point of the grid that contains two active elements at distance smaller than λ/2, so that r_4>1, this making the whole layout superdirective, *i.e.*, not belonging to the set of possible solutions to problem (2.2) (this is why it has been made superdirective on purpose).

The definition and the use of this matrix is one of the most innovative contributions of this thesis work, since, to the best of the student's knowledge, it has never been applied before our publications [48], [99].

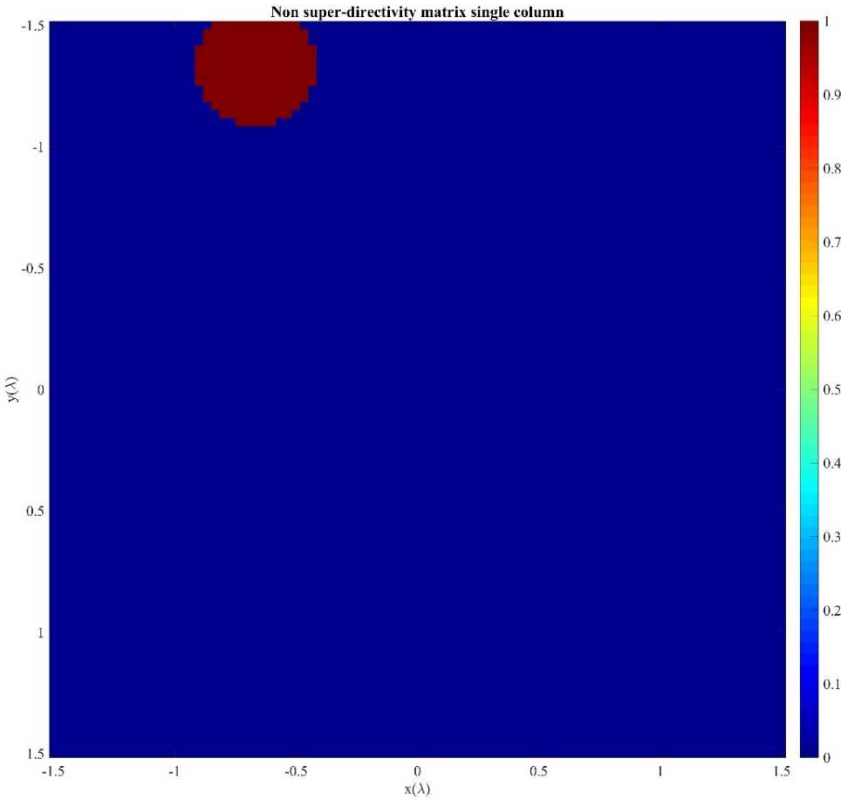

Figure 19. Single column of matrix **A**. The red points represent all the points at distance smaller than half-wavelength with respect to the circle's center, whereas the blue ones represent all the points at distance greater than or equal half-wavelength

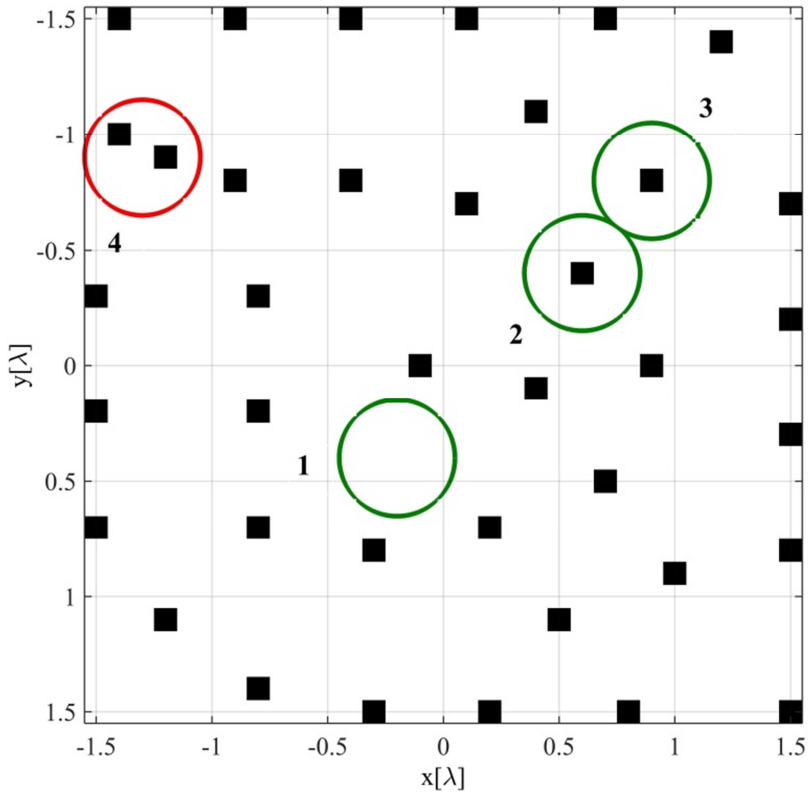

Figure 20. Example of four circles, related to four specific columns of **A**, on a general array layout identified by $\tilde{\mathbf{w}}_b$.

3.3.5 The shared layout

The condition $|\mathbf{w}| \leq \mathbf{w}_b$ (which has already been utilized in [50] for instance) in (2.2) assures the layout in Tx and Rx is the same. In our case we only distinguish between Tx and Rx, but more in general this type of relation can be exploited for multiple patterns that are required to share the same layout (see relation (12) in [50] for more details).

3.3.6 The cost function

In (2.2) the cost function, that is equivalent to minimize the ℓ_0 pseudo-norm of vector **w** since minimization of this functional will result in the minimum number of nonzero weights, favors sparse arrays and is therefore particularly suited for the desired sparseness. Although the cost function is linear with respect to the variable $\mathbf{w}_b$ (it is just the minimization of its sum) it is not as such in functional terms because it is discontinuous and has zero gradient except at the discontinuities [92]. However, we can consider the relationship between maximally sparse optimization and the generalized l_p optimization for $0<p<1$ and show the conditions under which a solution to the computationally simpler l_p optimization problem leads to a solution of the cost function in (2.2). Let us consider the generalized optimization problem

$$\min_{\mathbf{w}\in\mathbb{C}^{NM}} g(\mathbf{w}) = \sum_{i=1}^{NM} |w_i|^p \tag{2.10}$$

for $0<p<1$. To show how this problem is related to the cost function in (2.2) let us consider the unit ball surfaces in $\mathbf{R}^2$ for the quasi-norm

$$\|\mathbf{w}\|_{l_p} = \left[\sum_{i=1}^{NM} |w_i|^p \right]^{(1/p)} \tag{2.11}$$

for values of p in the range $0 \leq p \leq \infty$ as illustrated in Figure 21.

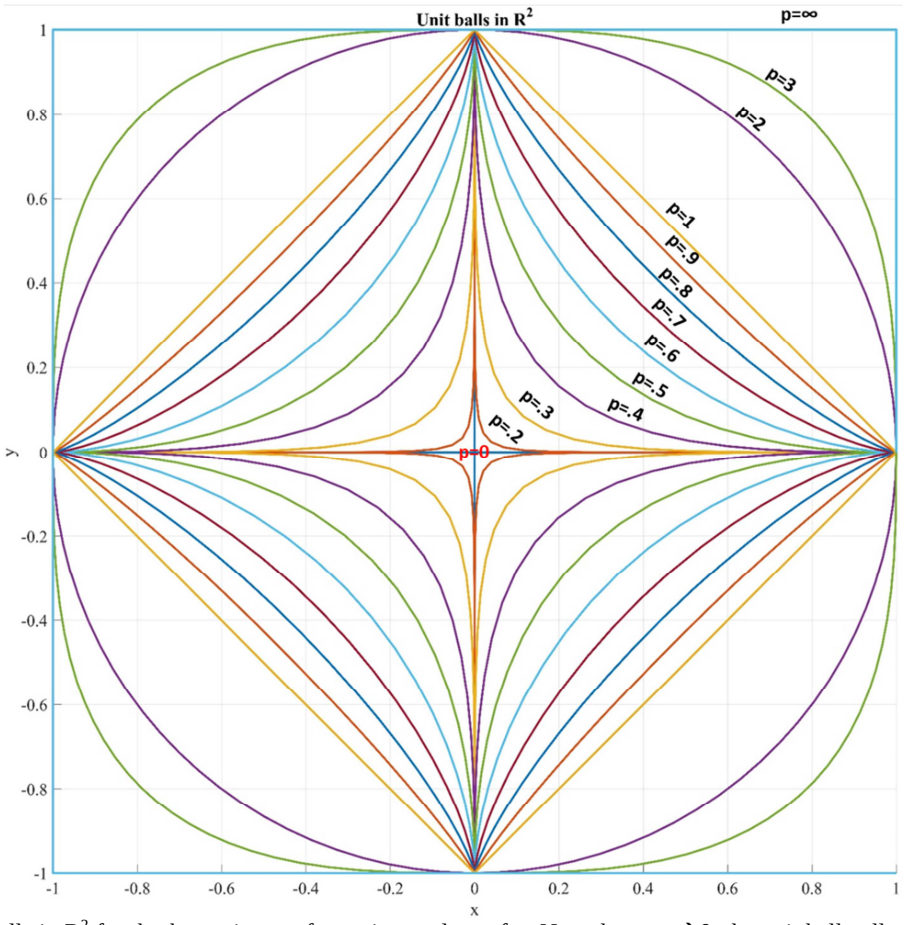

Figure 21. Unit balls in R^2 for the l_p quasi-norm for various values of p. Note that as $p \rightarrow 0$, the unit ball collapses onto the (x,y) axes (the case $p=0$, highlighted in red, is the binary case "*x or y*").

For $p \geq 1$ we have the conventional norm defined by (1.39), which is a convex functional and obeys the triangle inequality. An example of convex geometry is depicted in Figure 22: the unit ball in $\mathbf{R}^3$ for $p=3$.

For $0<p<1$ l_p is only a quasi-norm [93] since the triangle inequality does not hold. Over $\mathbf{R}^{NM}$ $\|\mathbf{w}\|_{l_p}$ is neither convex nor concave, containing many strong local minima and presenting a difficult optimization problem. Large values of p result in smooth solutions, but as $p \rightarrow 0$ they tend to become more "spiked", or sparse [92].

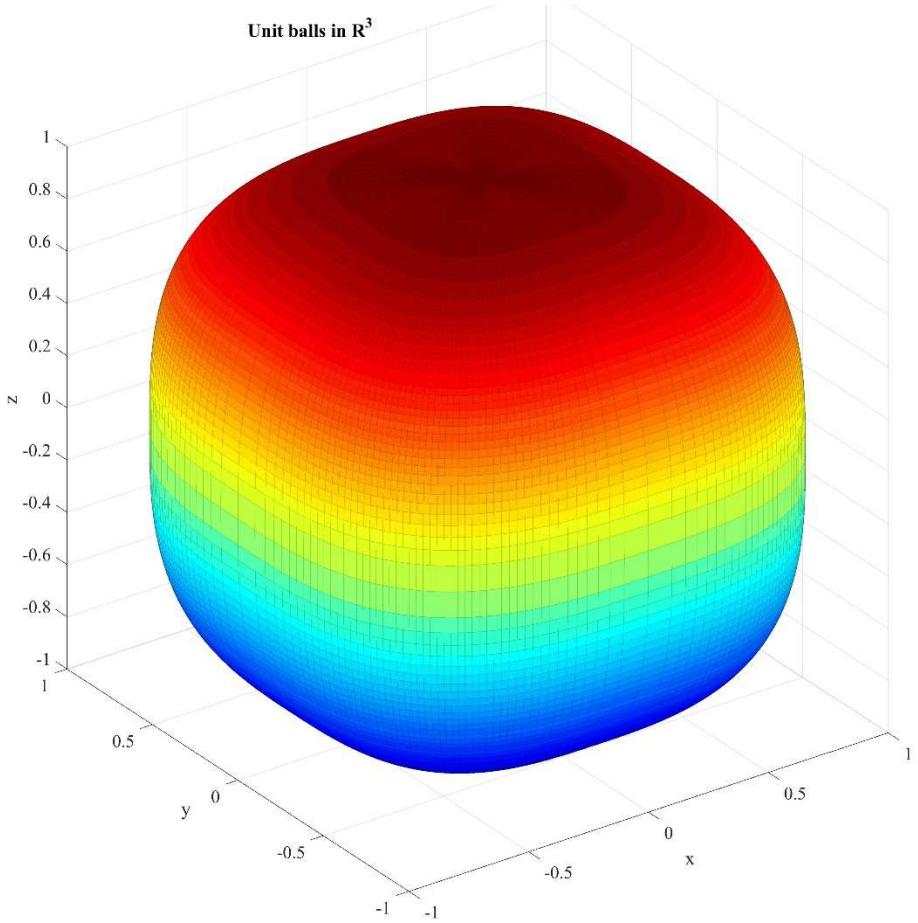

Figure 22. Example of convex geometry: unit ball in R^3 for the l_p norm for $p=3$.

An example of concave geometry is given in Figure 23, where the aforementioned "spiked" solution is clear: the unit ball in R^3 for p=.5. For p=0 the unit ball in Figure 21 lies on the (x,y) axes. We observe, in general, that

$$\lim_{p\to 0}|x|^p = \begin{cases} 1, & for\ x \neq 0 \\ 0, & for\ x = 0 \end{cases} \tag{2.12},$$

which suggests that we may identify minimum order optimization as a special case of generalized l_p optimization:

$$\lim_{p\to 0}\left[g(\mathbf{w})\right]^{\frac{1}{p}} = \lim_{p\to 0}\left[\sum_{i=1}^{NM}|w_i|^p\right]^{\frac{1}{p}} = \|\mathbf{w}\|_{l_0} = \|\mathbf{w}\|_0 \tag{2.13}.$$

The utility of this observation is that for p>0 $g(\boldsymbol{w})$ eliminates some of the handicaps of $\|\mathbf{w}\|_0$: it is indeed continuous everywhere and differentiable except at the axes. Theorem 4 in [92] provides justification for minimum order optimization based on minimizing $g(\boldsymbol{w})$ by proving that for a bounded basic feasible solution set there exists a finite p_1>0 such that for all 0<p<p_1, any solution to (2.10) is a solution to the cost function of (2.2) [92].

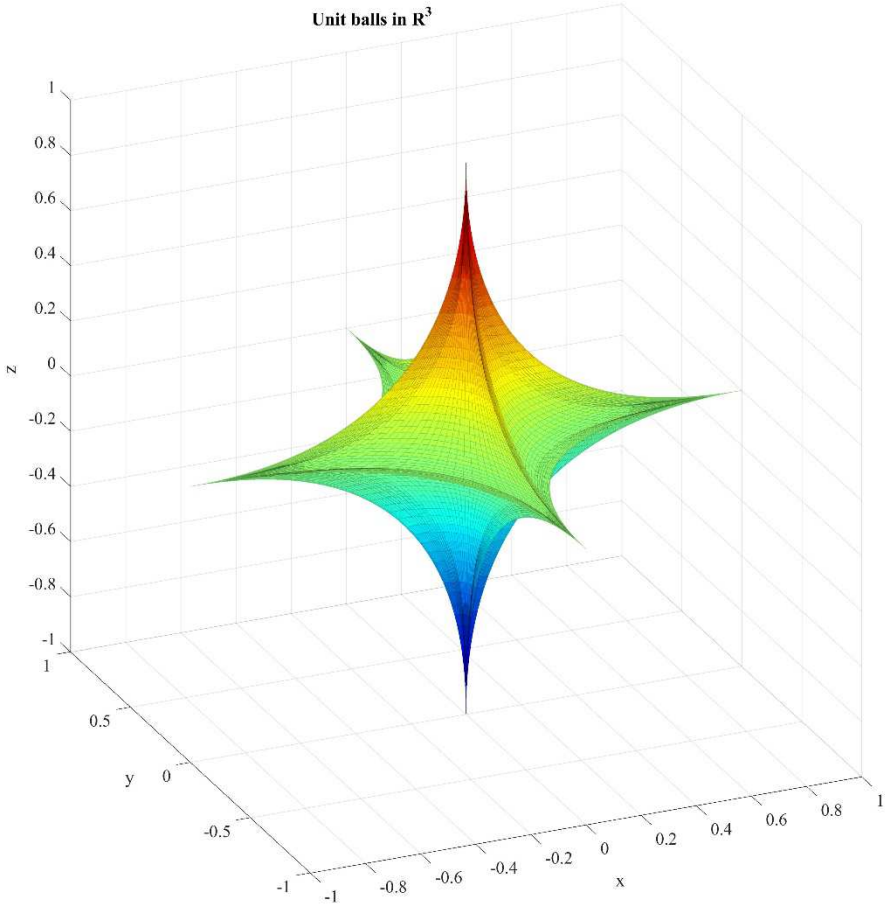

Figure 23. Example of concave geometry: unit ball in R^3 for the l_p quasi-norm for $p=0.5$.

The optimization problem (2.10) therefore defines a class of problems, indexed by p, whose global solutions are increasingly sparse as p decreases, until $p<p_1$, at which point an optimally sparse solution is given. As p varies over the range $0<p<1$ only a finite number of unique globally optimum solutions are encountered [94]. A value $\mathbf{w}_{opt}$ will remain as such over a range of p values, and as p decreases, we step from one solution to another in a discrete fashion [94]. Theorem 4 in [92] proves that (global minimum) solutions to (2.10), for $p<p_1$, form a subset of (global minimum) solutions to the cost function of (2.2), so that any l_p optimum is acceptable until $0<p<p_1$. Problem (2.10) is closely related to the problem of concave minimization over linear constraints. One of the techniques proposed for solving this type of minimization is the Branch-and-Bound (B&B) procedure [92], which we are giving a dedicated section.

3.4 Problem Solution: reweighted ℓ_1-norm and B&B algorithms

Hereinafter we are proposing two different approaches for solving the MIP binary problem (2.2): the former is a method substantially based on the solution of (2.2) by means of a fast B&B algorithm [62[67], which solves a set of convex problems in parallel. The latter is a method based on the sequential convex minimization of the weighted ℓ_1-norm based, in turn, on the minimization of a convex measure of the ℓ_0–norm [73] by using an interior point algorithm [55].
We have experienced much shorter simulation times by using an optimization technique based on the reweighted ℓ_1-norm with respect to the use of global methods, also in case of a small number of variables. Therefore, although global stochastic optimization methods like, for instance, SA and GA, could be used as well, the computational load is prohibitive for the goals we set, and besides the cost function is upper and lower bounded (neither the array can have more elements than its full version nor it can have less than one element, respectively), which turns out to be quite helpful for the B&B. Moreover, differently from global stochastic methods, the latter returns an exact solution (deterministic approach), and in any case its efficiency depends on how it is implemented.

3.5 B&B algorithms

Hereinafter it is described how the general B&B algorithm works in its classical formulation together with its main aspects and some examples as well. This section does not mean to cover this type of algorithm exhaustively as is currently done in mathematics/computer science research. At present, much more advanced versions [63] of the basic B&B algorithm herein discussed are normally employed for solving MIP problems.

B&B is a general-purpose approach capable of solving pure IP, mixed IP, and binary IP problems. We assume in the following that the given problem is a maximization one, since modification of the algorithm for the minimization problem is straightforward [57]. Theoretically, any pure IP problem with finite bounds on integer variables can be solved by enumerating all possible combinations of integer values and determining a combination (solution) that satisfies all constraints and yields the maximal objective value (hence the name of complete enumeration). Unfortunately, the number of all possible combinations is prohibitively large to be evaluated even for a small problem. As a better alternative, implicit enumeration applies an intelligent enumeration scheme that can cover all possible solutions by explicitly evaluating only a small number of them while ignoring (or implicitly enumerating) a large number of inferior solutions. One such strategy is called *divide and conquer*. Basically, this strategy divides the given problem into a series of easier to solve subproblems that are systematically generated and solved (or conquered). The solutions of these generated subproblems are then put together to solve the original problem. B&B can be viewed as a divide and conquer approach to solving the IP problem, in which a branching process is for dividing and a bounding one for conquering. As the algorithm proceeds, a series of Linear Programming (LP) subproblems are systematically generated and solved. Then the upper and lower bounds are progressively tightened on the objective value of the original IP problem. A typical way to represent such process is via the B&B tree, which is a specialized enumeration tree for keeping track of how LP subproblems are generated and solved. The root node that represents the LP relaxation of the original IP problem is solved. If the LP optimum solution satisfies the integer requirement then the IP problem is solved. Otherwise, the LP objective value becomes the initial upper bound on the IP optimal objective value and the root node is partitioned into two successor nodes (subproblems) by two branches. These branches have the following properties:

a) they cut off current non-integer LP optimum point and other fractional region;
b) the two successor nodes are mutually exclusive and their union contains the same integer feasible region as that of their predecessor (*i.e.*, no integer points are eliminated).

The solution of an LP relaxation on a node provides information about

a) whether a further branching from this node is needed (or whether the node can be pruned), and
b) a better lower bound (for maximization problem) on the objective of the original IP problem.

There are three cases indicating that a node can be pruned:

1. the subproblem has no feasible LP solution;
2. the subproblem has an integer optimum solution, and
3. the upper bound of the subproblem optimum is less than or equal to the lower bound of the original problem.

These three cases are, respectively, referred to as *pruned by infeasibility*, *pruned by optimality* and *pruned by bound*. If a node is pruned by optimality its optimum solution can be used to increase the lower bound on the objective value of the original IP problem.

Whenever an integer solution to a subproblem is obtained, it is a *candidate optimum* to the original IP problem. In the solution process of B&B the best integer solution found so far is continuously updated. Such solution is called *incumbent*.

The B&B algorithm is usually depicted as an enumeration tree, in which the nodes denote the subproblems and the branches correspond to constraints (cuts) that separate the subproblems from their parent subproblems. The number above each node is the optimal solution to the LP subproblem generated on that node (which is also the upper bound on that branch). The number below the node indicates the best

lower bound on the original IP problem found so far. Examples of B&B trees are depicted in Figure 27, Figure 28 and Figure 29
When an LP solution contains several fractional variables, the decision of which integer variable to branch on next is needed. The following rules are commonly used for choosing a branching variable:

- Variable with fractional value closest to 0.5
- Variable with highest impact on the objective function
- Variable with the least index.

A decision is also needed as to which unpruned node to explore first. The most commonly used search strategies include

1. Depth-first (last-in first-out; solve the most recently generated subproblem first)
2. Best-bound-first (best upper bound; branch on the active node with greatest objective value)

The goal of the depth-first strategy is to quickly obtain a primal feasible integer solution whose objective function is a lower bound on the given IP problem and can be used to prune nodes by bound (rule 3). The best-bound-first strategy chooses the active node with the best upper bound (for maximization problem). The goal is to minimize the total number of nodes evaluated in the B&B tree. Performances of these branching rules depend on the problem structure. In practice, a compromise between the two is adopted. That is, apply the depth-first strategy to first get one feasible integer solution, followed by a mixture of either strategies.

Now we can describe the general B&B algorithm using the following notation.

S= the given IP problem
S_{LP}= the LP relaxation of S
y_{LP}= the solution to the LP relaxation of the given IP problem
$z^* = f(y_1, y_2, \ldots, y_j, \ldots, y_K)$ = the objective function of the K variables $(y_1, y_2, \ldots, y_j, \ldots, y_K)$
$\bar{z}$ = lowest (best) upper bound on z^* of the given IP problem
$\underline{z}$ = highest (best) lower bound on z^* of the given IP problem

These are global bounds that are periodically updated as the branching proceeds down the various paths in the tree, but are not shown on the latter. Next comes further notation.

S^k= subproblem k of problem S
S_{LP}^k = the LP relaxation of subproblem k
z^k = the optimum objective value of S^k
$\bar{z}^k$ = lowest (best) upper bound of subproblem S^k (shown above node k)
$\underline{z}^k$ = highest (best) lower bound of subproblem S^k (shown below node k)
y_{LP}^k = the optimum solution of the LP subproblem S_{LP}^k
$\bar{y}_j$ = non-integer value of integer variable y_j (current numerical value of y_j)
$\lfloor a \rfloor$ = the largest integer ≤a (or rounding down a)
$\lceil a \rceil$ = the smallest integer ≥a (or rounding up a)

Now we can formally describe the B&B procedure.

Step 0 (Initialization). Solve the LP relaxation (S_{LP}) of the given IP problem (S). If it is unfeasible so is the IP problem, thus terminate. If the LP optimum solution satisfies the integer requirement the IP

problem is solved, thus terminate. Otherwise, initialize the best upper bound ($\bar{z}$) by the optimal objective value of problem S_{LP} and the best lower bound by $\underline{z} = -\infty$. Place S_{LP}^{k} on the active list of nodes (subproblems). Initially, there is no incumbent solution.

Step 1 (Choosing a node). If the active list is empty, terminate. The incumbent solution y* is optimal. Otherwise choose a node (subproblem) S^k with S_{LP}^{k} as LP relaxation, by one of the aforementioned rules (*i.e.* depth-first, best-bound-first, etc.).

Step 2 (Updating Upper Bound). Solve and set $\bar{z}^k$ equal to the LP optimum objective value. Keep the optimum LP solution y_{LP}^{k} .

Step 3 (Prune by infeasibility). If S_{LP}^{k} has no feasible solution, prune the current node and go to step 1. Otherwise go to step 4.

Step 4 (Prune by Bound). If $z^k \le \underline{z}$ prune the current node and go to step 1. Otherwise go to step 5.

Step 5 (Updating Lower Bound and Pruning by optimality).

a) If the LP optimum y_{LP}^{k} is integer a feasible solution S is found, and also an incumbent solution to the given problem. Set $\underline{z}^k = y_{LP}^{k}$ and compare $\underline{z}^k$ with $\underline{z}$. If $\underline{z}^k \ge \underline{z}$ set $\underline{z}^k = \underline{z}$, otherwise $\underline{z}$ does not change. The current node is pruned because no better solution can be branched down from it. Go to step 1.
b) If the LP optimum y_{LP}^{k} is non-integer, go to step 6.

Step 6 (Branching). From the current node S^k choose a variable y_j with fractional value to generate two subproblems S_1^K and S_2^K defined by

$$\begin{aligned} S_1^K &= S^k \cap \left\{ y : y_j \le \lfloor \bar{y}_j \rfloor \right\} \\ S_2^K &= S^k \cap \left\{ y : y_j \ge \lceil \bar{y}_j \rceil \right\} \end{aligned} \qquad (2.14).$$

Place both these two nodes in the active list and go to step 1.

3.5.1 An example of IP problem

Solve the following pure IP problem by using the B&B algorithm:

$$\begin{aligned} &Maximize\ z = 5y_1 - 2y_2 \\ &s.t. \\ &-y_1 + 2y_2 \le 5 \\ &3y_1 + 2y_2 \le 19 \\ &y_1 + 3y_2 \ge 9 \\ &y_1, y_2 \ge 0 \\ &y_1, y_2 \in \mathbb{N} \end{aligned} \qquad (2.15)$$

We first solve the LP relaxation S_{LP}. As shown in Figure 24, the shaded area represents the LP feasible region, and the solid lattice points the IP feasible region.

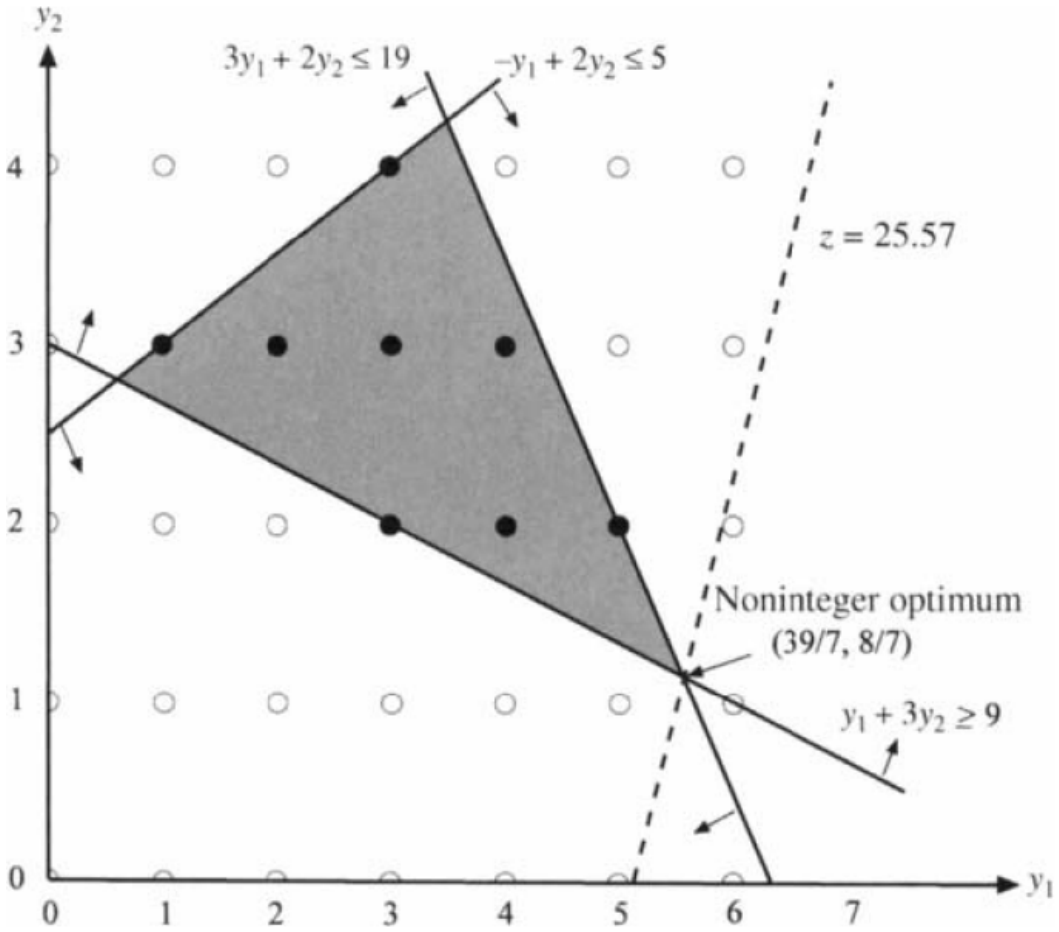

Figure 24. LP and IP feasible regions

We obtain the noninteger optimum y_1=39/7, y_2=8/7, and z=25.57. Then the objective value 25.57 becomes an upper bound to the IP problem. At this point we set the lower bound to -∞. Since both y_1 and y_2 are fractional we need to branch on them in an attempt to obtain an integer optimum. We arbitrarily select y_1 as the variable to be branched. Two subproblems are generated by adding the constraints $y_1 \leq 5$ and $y_1 \geq 6$ to the LP relaxation. From Figure 25 we can see that the triangle area S' is cut off by $y_1 \leq 5$. Clearly the branch with the added constraint $y_1 \geq 6$ is infeasible, thus it is pruned by infeasibility. The other branch with the added constraint $y_1 \leq 5$ is optimized at $(y_1,y_2)=(5,4/3)$, with objective value 22.33. So the new upper bound is updated to 22.33. Again, the variable y_2 is fractional, so this time we branch on y_2. The two constraints $y_2 \geq 2$ and $y_2 \leq 1$ are then added. This time the area S'' is cut off, as shown in Figure 26. The branch with $y_2 \leq 1$ is infeasible and hence is pruned by infeasibility. The one with $y_2 \geq 2$ is optimized at $(y_1,y_2)=(5,2)$, with objective value 21. Since this is a feasible solution to the IP problem the value 21 becomes a new lower bound to the problem, replacing -∞, and (5,2) is a candidate solution. Checking the tree, all branches are evaluated, so $(y_1,y_2)=(5,2)$ is the optimal solution to the IP problem, and the optimal objective value is 21. The enumeration tree for this example is in Figure 27. We observe that if at the beginning we branched on y_2=8/7 we would get to the final result earlier, thus the adoption of a decision criterion on which node to explore first is advisable.

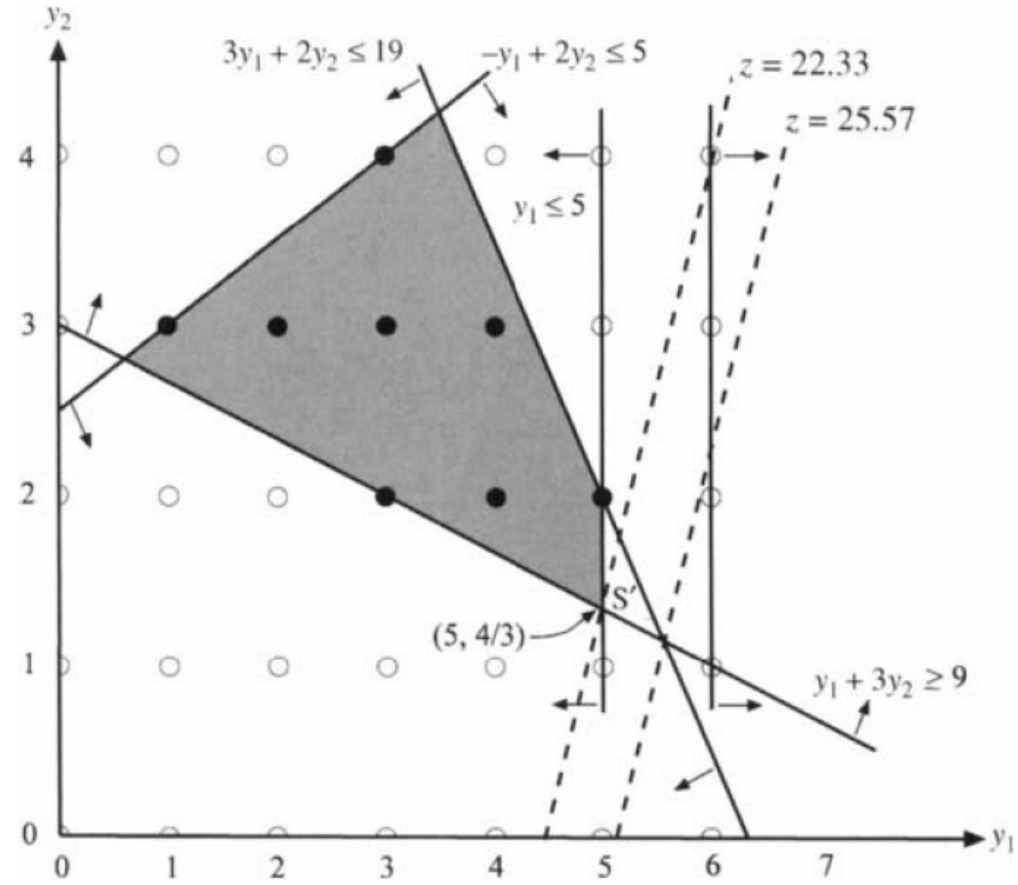

Figure 25. LP and IP feasible regions after the first branching.

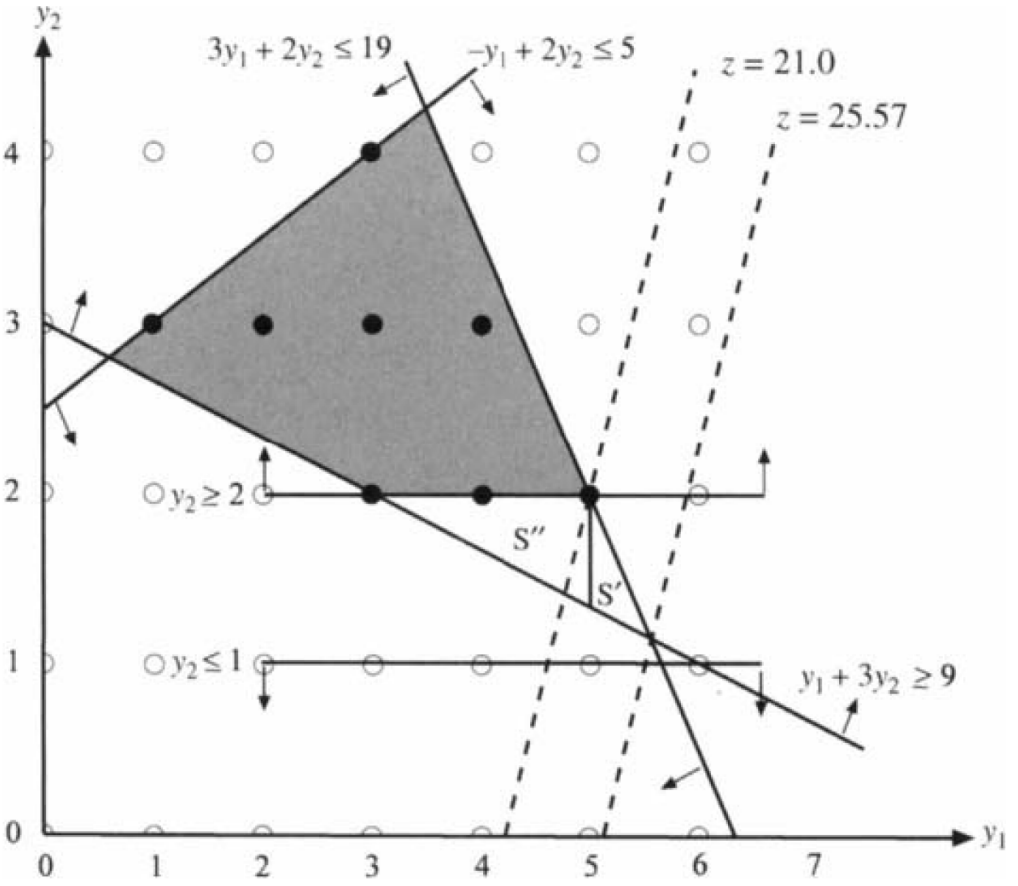

Figure 26. LP and IP solution regions after the second branching.

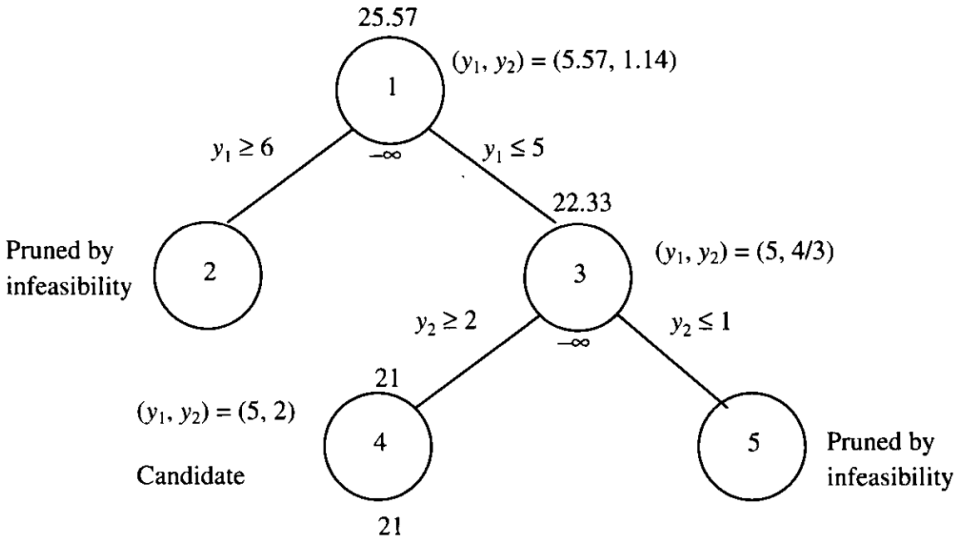

Figure 27. B&B tree for problem (2.15).

3.5.2 An example of IP problem with applied criteria

Solve the following mixed integer problem using B&B approach and applying the rule of best-bound-first at each step. Furthermore, at each node select the variable with the least index to branch on first.

$$
\begin{aligned}
&\textit{Maximize } z = -y_1 + 2y_2 + y_3 + 2x_1 \\
&\textit{s.t.} \\
&y_1 + y_2 - y_3 + 3x_1 \leq 7 \\
&3y_3 + y_2 - x_1 \leq 5 \\
&3y_1 + x_1 \geq 2 \\
&y_1, y_2, y_3 \geq 0 \\
&x_1 \geq 0 \\
&y_1, y_2, y_3 \in \mathbb{N}
\end{aligned}
\qquad (2.16).
$$

After solving the LP relaxation we obtain an LP optimum y_1=6/11 (which is, in this case, the integer variable with least index), y_2=59/11, y_3=0, x_1=4/11 and z=120/11≅10.91. We use this solution as the root node of the B&B tree in Figure 28, in which the number of each node indicates the sequence of subproblems evaluated. Note that at node 1 the constraint $y_1 \leq 0$ was indicated on the left branch, but since from (2.16) the relation $y_1 \geq 0$ must hold, y_1 has to be fixed at 0. At node 7, the constraint $y_2 \geq 2$ was intended to be added, but if we trace back along node 7, we see that the constraint $y_2 \leq 2$ was already added at node 2. Combining these two constraints, we have y_2=2. Same applies to the constraint of y_3=2 at node 8. The problem is finally optimized at node 12, where (1,5,0,1/3) is the optimal solution, with objective value 9.67. Figure 29 depicts the B&B tree for the same problem where the depth-first rule is applied, and at each node the variable (violating an integer constraint) with the largest absolute value cost coefficient is chosen to branch on first. Depth-first is sometimes called Last-In First-Out (LIFO) because it solves the most recently generated subproblem first. It tends to pursue paths to the depths of the tree, then backtrack to where that path started, and finally plunge down into another depth search. Therefore another name for depth-first is “backtracking”. Best-bound-first is sometimes called “jumptracking” because it leads to searches that jump back and forth across the tree.

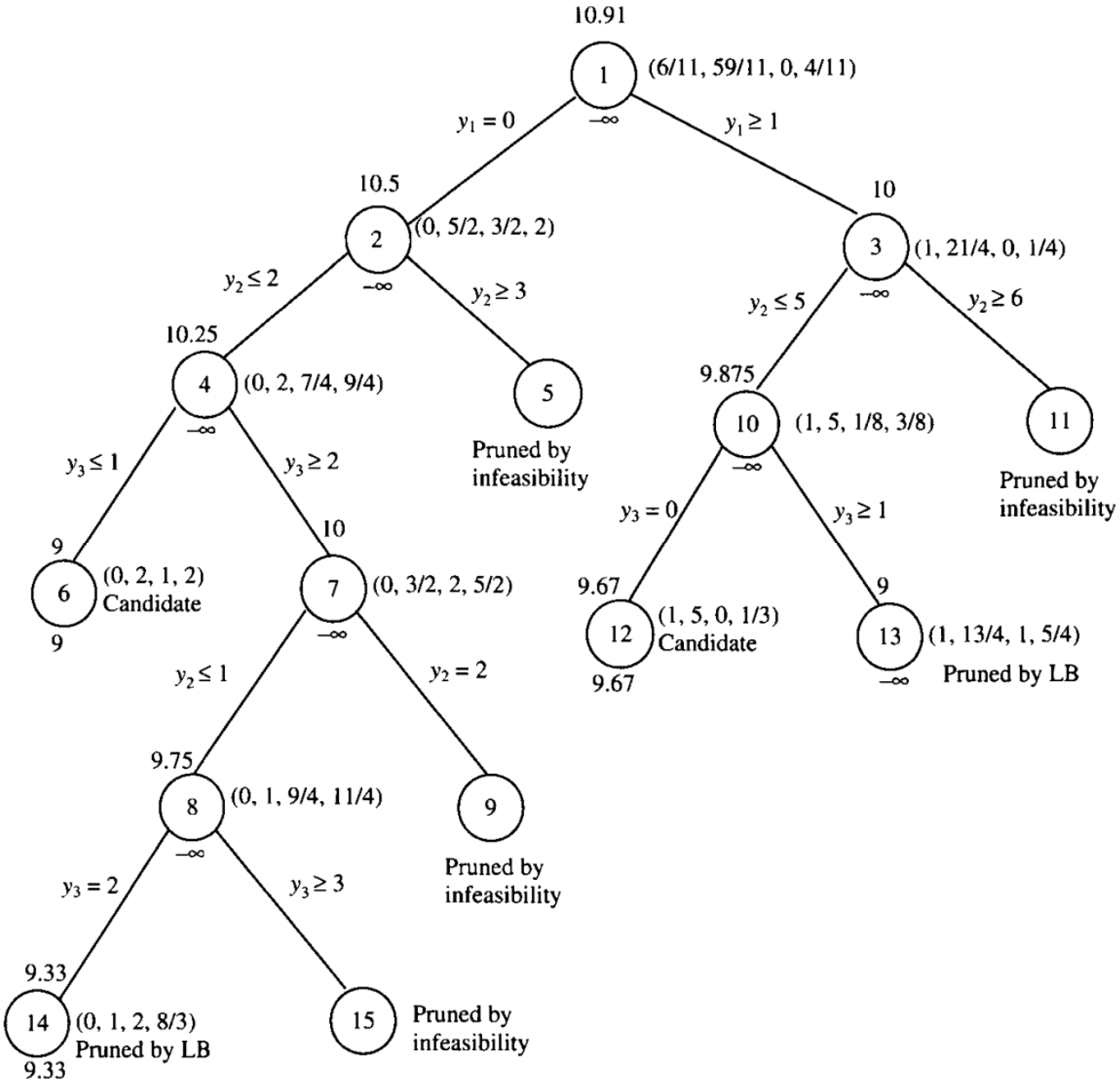

Figure 28. B&B tree for problem (2.16) using best-bound-first

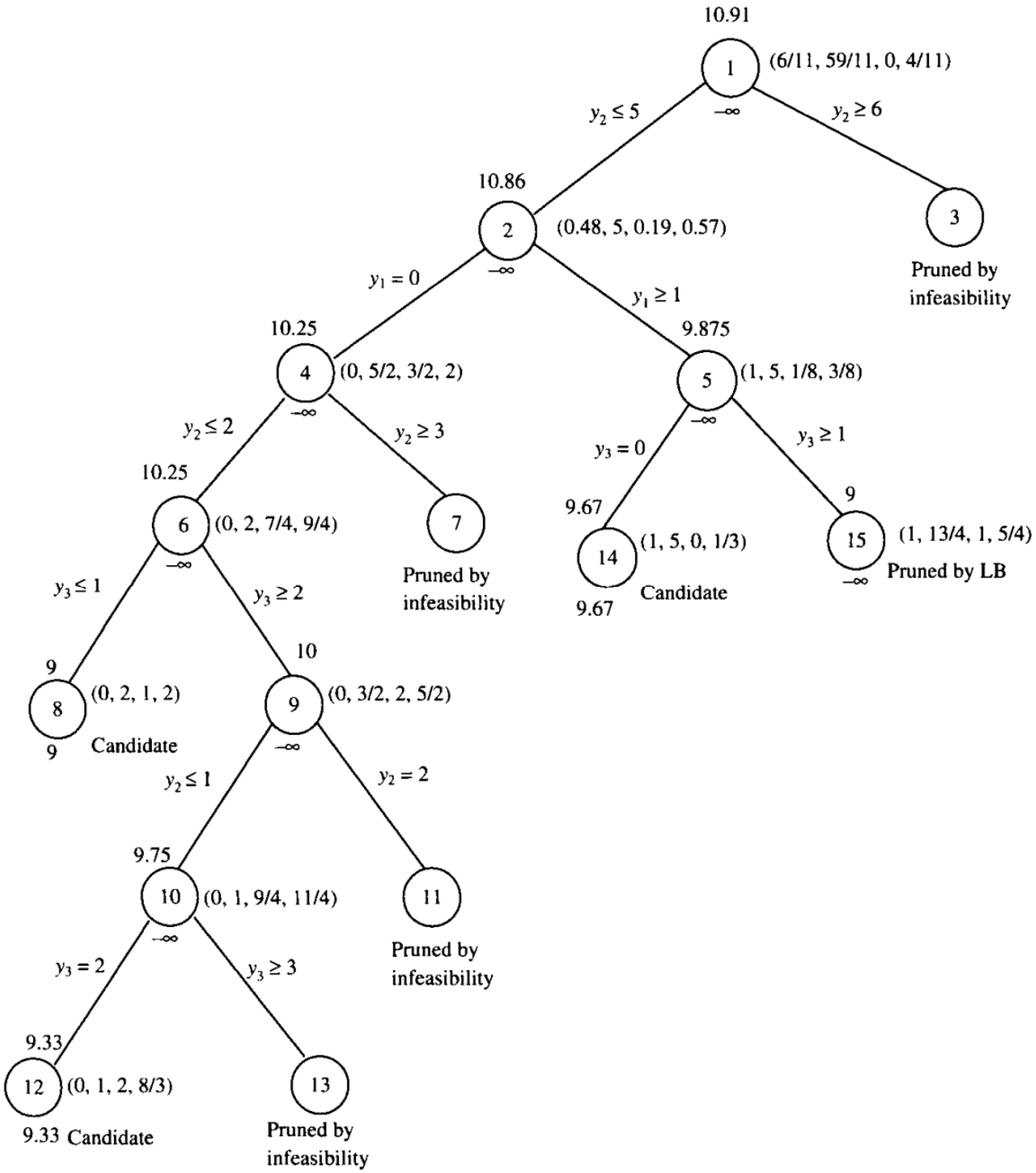

Figure 29. B&B tree for problem (2.16) using depth-first

We end this section by stating that the B&B algorithms can use multiple cores (of multiple machines) during the optimization process; in principle, if one had as many cores as the potentially required nodes of the B&B tree then a core could be assigned to each subproblem, leading in this case to a full parallelization. In practice clearly, an ensemble of subproblems is assigned to each core.

3.6 Reweighted ℓ_1-norm

A common alternative to relation (2.3) is to consider the ℓ_1-norm of vector ***w***, which is indeed known to produce sparse solutions for a wide range of applications [95[96]. In the ambit of compressive sensing for instance one wishes to recover a sparse signal $\mathbf{x}_0 \in \mathbb{R}^n$ from a small number of measurements $\mathbf{y} = \mathbf{\Phi x}_0$, where $\mathbf{\Phi}$ is a *m* x *n* matrix whose *m* columns (with $m \ll n$) represent the measurements on the signal. It has been shown that ℓ_1 minimization allows recovery of sparse signals from remarkably few measurements [97[98]: supposing $\mathbf{\Phi}$ is chosen randomly from a suitable distribution, then with very high probability, all sparse signals $\mathbf{x}_0$ for which $\|\mathbf{x}_0\|_0 \le m/\alpha$ with $\alpha = O[\log(m/n)]$ can be perfectly recovered by using the ℓ_1 minimization. However, a key difference between ℓ_0 and ℓ_1 norms lies in the dependence on magnitude: larger coefficients are penalized more heavily in the ℓ_1 norm than smaller ones, unlike the more democratic penalization of the ℓ_0 norm. Therefore, to further enhance the sparsity of the solution an algorithm that consists in solving a sequence of weighted ℓ_1 minimization problems, where the weights used for the next iteration are computed from the value of the current solution, has been proposed in [73]. The idea of this algorithm is to bring the ℓ_1 criterion as close as possible to the ℓ_0 one, penalizing more democratically the nonzero coefficients. The algorithm proposed by E.J. Candes, M. B. Wakin and S. Boyd has been developed with the goal of reconstructing sparse signals from highly incomplete sets of measurements. Although not developed specifically for the synthesis of sparse arrays, the procedure described in [73] can be efficiently exploited for our scope. In fact the measurements one would use in the recovery of a sparse signal are substituted by FF specifications, while the array layout is actually the "sparse signal" to be recovered. Consider the weighted ℓ_1 minimization problem

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{i=1} w_i |x_i| \\ &s.t. \\ &\mathbf{y} = \mathbf{\Phi x} \end{aligned} \qquad (2.17),$$

where $w_1, \ldots, w_n$ are positive weights and $\mathbf{\Phi}$ is an *m x n* matrix with, typically, $m<n$. It is convenient to denote the objective functional by $\|\mathbf{Wx}\|_1$, where $\mathbf{W}$ is a diagonal matrix with $w_1, \ldots, w_n$ on its diagonal and zero elsewhere. One possible use for the weights could be to counteract the influence of the signal magnitude on the ℓ_1 penalty function. Suppose for example that the weights are inversely proportional to the true signal magnitude, *i.e.* that

$$w_i = \begin{cases} \dfrac{1}{|x_{0,i}|} & x_{0,i} \neq 0 \\ \infty & x_{0,i} = 0 \end{cases} \qquad (2.18).$$

If the true signal $\mathbf{x}_0$ is k-sparse, i.e. $\|\mathbf{x}_0\|_0 \le k$, then (2.17) is guaranteed to find the correct solution with this choice of weights provided that $m \ge k$ and that the columns of $\mathbf{\Phi}$ are in general positions. The large (actually infinite) entries of in w_i force the solution $\mathbf{x}$ to concentrate on the indices where w_i is small (actually finite), and by construction these correspond precisely to the indices where $\mathbf{x}_0$ is nonzero. Without knowing the signal $\mathbf{x}_0$ itself it is impossible to construct the precise weights (2.18), but this suggests more generally that large weights could be used to discourage nonzero entries in the recovered signal, while small ones could be used to encourage nonzero entries. Specifically, the algorithm is as follows:

1. Set the iteration count *l* to zero and $w_i^{(0)} = 1, \; i = 1, \ldots, n$.
2. Solve the weighted ℓ_1 minimization problem

$$x^{(l)} = \arg\min \left\| \mathbf{W}^{(l)} \mathbf{x} \right\|_1$$
$$s.t. \qquad (2.19).$$
$$\mathbf{y} = \mathbf{\Phi x}$$

3. Update the weights: for each *i=1,…,n,*

$$w_i^{(l+1)} = \frac{1}{\left| x_i^{(l)} \right| + \varepsilon} \qquad (2.20).$$

4. Terminate on convergence or when l attains a specified maximum number of iterations $l_{\max}$. Otherwise, increment l and go to step 2.

The parameter $\varepsilon>0$ in (2.20) is used to provide stability and to ensure, in particular, that a zero-valued component in $x^{(l)}$ does not strictly prohibit a nonzero estimate at the next step. The parameter ε should be set slightly smaller than the expected nonzero magnitudes of $\mathbf{x}_0$. The use of an iterative algorithm to construct the weights w_i tends to allow for successively better estimation of the nonzero coefficient locations. Even though the early iterations may find inaccurate signal estimates, the largest signal coefficients are most likely to be identified as nonzero. Once such locations are identified, their influence is downweighted in order to allow more sensitivity for identifying the remaining small but nonzero signal coefficients.
The reason why the reweighted ℓ_1 minimization can improve the recovery of sparse signals lies in the connection it has with the following problem:

$$\min_v g(v)$$
$$s.t. \qquad (2.21),$$
$$v \in \Theta$$

where Θ is a convex set. In (2.21) g is concave [73], hence one can improve on a guess v at the solution by minimizing a linearization of g around v. This simple observation yields a Majorization-Minimization (MM) algorithm whose each iteration is the solution to a convex optimization problem [73] which turns out to be equivalent to problem (2.19). A particular case of (2.21) is the log-sum penalty function:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \sum_{i=1}^{n} \log\left(\left| x_i \right| + \varepsilon \right)$$
$$s.t. \qquad (2.22),$$
$$\mathbf{y} = \mathbf{\Phi x}$$

which has the potential to be much more sparsity-encouraging than the ℓ_1 norm. Let us consider indeed three potential penalty functions for scalar magnitudes t:

$$f_0(t) = 1_{\{t \neq 0\}}, \quad f_1(t) = |t|, \quad f_{\log,\varepsilon}(t) \propto \log\left(1 + \frac{|t|}{\varepsilon} \right),$$

where in the latter the constant of proportionality is set such that $f_{\log,\varepsilon}(1) = 1 = f_0(1) = f_1(1)$, see Figure 30. The first (ℓ_0-like) penalty function f_0 has infinite slope at $t=0$, while its convex (ℓ_1-like) relaxation f_1 has unit slope at the origin. The concave penalty function $f_{log,\varepsilon}(t)$, however, has slope at the origin that grows roughly as $1/\varepsilon$ when $\varepsilon \rightarrow 0$. Following this argument it would appear that ε should be set arbitrarily small in order to most closely make the log-sum penalty resemble the ℓ_0 norm. Unfortunately, as $\varepsilon \rightarrow 0$ it becomes more likely that the iterative reweighted ℓ_1 algorithm will get stuck in an undesirable local minimum [73]. A choice of ε slightly smaller than the expected nonzero magnitudes of $\mathbf{x}$ provides the stability

necessary to correct for inaccurate coefficient estimates while still improving upon the unweighted ℓ_1 algorithm for sparse recovery.

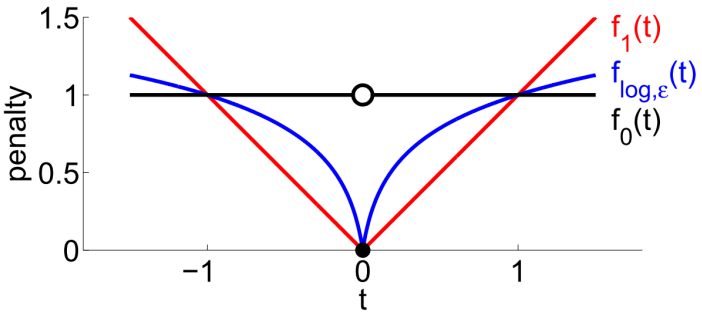

Figure 30. ℓ_0-like functions. At the origin, the canonical ℓ_0 sparsity count $f_0(t)$ is better approximated by the log-sum penalty function $f_{log,\varepsilon}(t)$ than by the traditional convex ℓ_1 relaxation $f_1(t)$.

3.7 The dynamics

For the Tx synthesis not only we seek for an as sparse as possible solution, but also for an ispohoric one, this implying that it must attain 0dB of dynamics (the ratio of the maximum to the minimum nonzero entry is 1). The reweighted ℓ_1-norm could be inappropriate for such scope, since every set of weights obtained through this algorithm will exhibit a dynamics different from 0dB. Even if the dynamics is quite narrow if one forces the nonzero entries of the obtained vector to be all equal the requirements could no longer be met. In general, when using the reweighted ℓ_1-norm in the attempt to approximate an isophoric solution, we typically experienced worse performances than with the MIP approach, but anyway to the best of the student's knowledge there is no known theoretical aspect that prevents a priori to achieve an equivalent solution. In general, the reweighted ℓ_1-norm enhances sparsity in terms of "zero" entries of the vector of interest, but not (necessarily) the homogeneity of magnitude in its "non zero entries", which is instead absolutely mandatory for Tx arrays. All these aspects set up the reason why we used either MIP and the reweighted ℓ_1-norm for synthesizing the arrays for the benchmark comparisons, but not for the synthesis of the sparse version of the chosen antenna, see 5.2. Hence in the following we are recasting problem (2.2) by means of the reweighted ℓ_1-norm with a focus on the Rx synthesis only (the constraints on upper and lower bounds for the Tx case are in fact absent). At the y^{th} iteration the problem to solve is:

$$\begin{aligned}
&\min_{\mathbf{w}_c \in \mathbb{R}^{NM}} \left\| \mathbf{Z}^{(y)} \mathbf{w}_c \right\|_1 \\
&s.t. \quad AF_{R_x}(u_0, v_0) = b \\
&\quad LB_h(u,v) \le \left| AF_{Rx}(u,v) \right|_{(u,v)\in\Phi_h} \le UB_h(u,v) \\
&\quad h = 1, \ldots, N_{R_x} \\
&\quad \mathbf{0} \le \mathbf{w}_c \le \mathbf{1} \\
&\quad |\mathbf{w}| \le \mathbf{w}_c \\
&\quad \mathbf{A}\mathbf{w}_c \le \mathbf{1}
\end{aligned} \qquad (2.23),$$

where $\mathbf{w}_c$ is an NM vector of continuous variables (whose entries are $w_c(n) \in [0,1]$, with n=1,2,…,NM) that represents the continuous version of vector $\mathbf{w}_b$ in (2.2), $\mathbf{Z}^{(0)} = \mathbf{I}_{NM}$ and $\mathbf{Z}^{(y)}$ is a diagonal matrix whose n^{th} element on its diagonal is computed as:

$$z_n^{(y)} = \frac{1}{w_{c_n}^{(y-1)} + \varepsilon}, \quad n \in \{1, 2, \ldots, NM\} \qquad (2.24),$$

b is the expected cardinality of the array, which at worst is equal to the one of the full-array, and obviously cannot be smaller than 1. In (2.24) the parameter ε is chosen as described earlier. The reason why in (2.23) either $\mathbf{w}_c$ and $\mathbf{w}$ are present although the problem is targeted to Rx patterns, lies in the fact that $\mathbf{w}_c$ is necessary for the non-superdirectivity constraint, which in this case is not guaranteed. Besides, the *b* parameter must be varied several times until the maximum sparse solution is reached. These are actually the drawbacks of this approach.

Chapter 4

Application of the developed algorithms

In this chapter we are going to apply the algorithms described in the previous ones either to several benchmarks of literature and in order to set new ones. As regards the literature benchmarks comparisons we have shown our algorithms perform better, meaning that they succeed in achieving greater reductions in the number of elements in some cases with even lower SLLs. As regards the new benchmarks we have derived how the reduction of the number of elements behaves with respect to the intervals of steering for different array sizes, and we have seen it is decreasing (confirming the intuition for which the less degrees of freedom the algorithms can use the less they can sparsify). The latter is another innovative contribution, since, to the best of our knowledge, it was never shown and confirmed before our publications [48] [99].

4.1 Introduction

In order to evaluate the capabilities and the indexes of performances of the herein proposed approaches, several benchmark problems are considered. In particular, a comparison with benchmark problems described in [46] and [50], concerning the synthesis of different types of power patterns are shown. Furthermore, a parametric study aimed at showing the comparison (in terms of magnitudes of sparsity) between square apertures of size 3λ, 5λ, and 7λ, for different intervals of steering and for different power masks, is discussed. In all the examples shown N_e represents the number of elements of the synthesized array. The quantities selected for the literature benchmark comparisons are the Q factor, verified according to [51] [100] and defined as the ratio between the stored to dissipated energy. It represents a robustness index in terms of bandwidth, efficiency, excitation sensitivity and position tolerances [51] [100]. A second index of performances is the ratio between the directivity of the synthesized array and the one pertaining to the isophoric case, confirming, obviously, for each case, that the former is non-superdirective as expected. Another index of performances is the aperture efficiency, defined, for hemispherically isotropic radiating elements, as

$$A_e = \frac{D}{4\pi A / \lambda^2} \tag{3.1},$$

where *D* and *A* are, respectively, the directivity and the area delimited by the convex hull of the synthesized array. Since in the following the directivity refers to the whole sphere of radiation, the quantity at the denominator of (3.1) is scaled by a factor of 0.5. Finally, we define the Elements Number Reduction Ratio (ENRR) as $1 - N_e / F$, which represents the measure of the achievable reduction of the active radiating elements with respect to the ones composing the full array. In such relation *F* is the number of elements of the full array.

4.2 Comparison with literature Benchmarks

Let us now consider the problem of synthesizing a sparse planar array of elements located on a square grid of side 5λ in order to radiate a pattern having a broadside main beam such that $\sin(\vartheta_{bw}) = 0.239$ at -6 dB. For such a problem, with the hybrid approach proposed in [17], a SLL below -16.5 dB is achieved with a planar sparse array composed of 41 elements. In [46] the same number of elements has been achieved, with a main BW of 0.24 at -6 dB and with a SLL of -17.3dB. By applying the proposed algorithms, the same BW and a maximum SLL of -17.5dB (that is optimum in a Chebyshev sense) has been obtained with a non-superdirective array layout, constituted by only N_e=35 elements. The optimized array layout is shown in Figure 31.

The Q factor and the directivity of the synthesized sparse array are, 3.79dB and 18.9dB respectively, whereas in the isophoric case they are 3.72dB and 19.1dB respectively (this confirms it is non-superdirective).

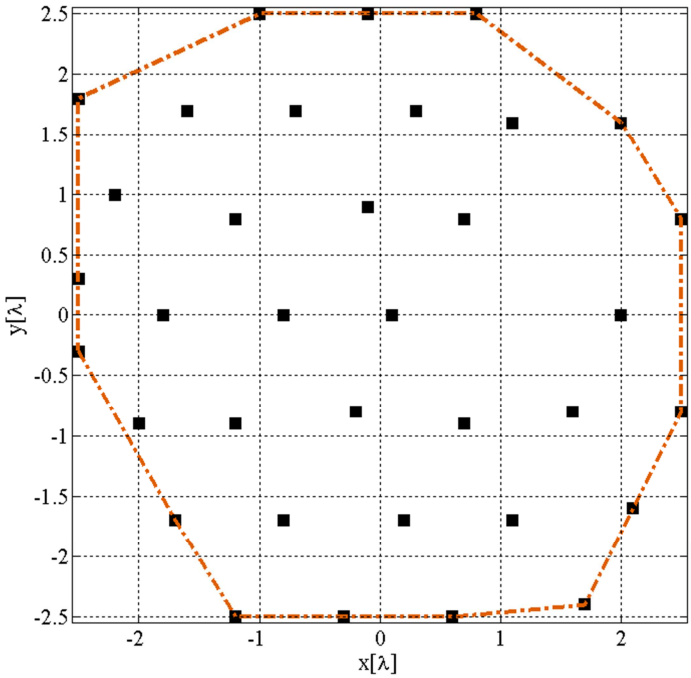

Figure 31. Resulting array layout relative to the synthesis of focused beam power pattern. Active radiating elements N_e=35. Discretization grid: $d_x=d_y=0.1\lambda$. The dashed curve represents the convex hull.

The area bounded by the convex hull is $21.11\lambda^2$, which causes a loss in aperture efficiency of 18.9dB-21.237dB=-2.337dB. In Figure 32 and in Figure 33 the 3D and 2D normalized synthesized power pattern, in the (u,v) domain, are shown respectively. The ϕ-cutting planes for ϕ=0°, 30°, 45°, and 90° are reported in Figure 34. The element positions and the associated normalized excitations, whose dynamics is 8.01dB, are reported in Table 2 .

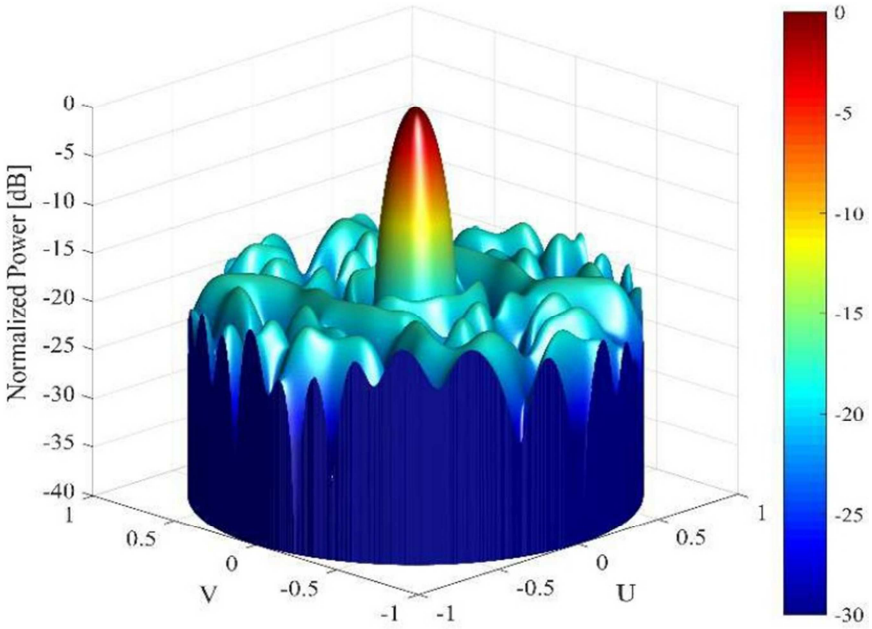

Figure 32. 3D normalized power pattern radiated by the array in Figure 31 with N_e=35 active radiating elements. BW= $\sin(\theta_{bw})$=0.240 at -6dB. SLL=-17.5dB.

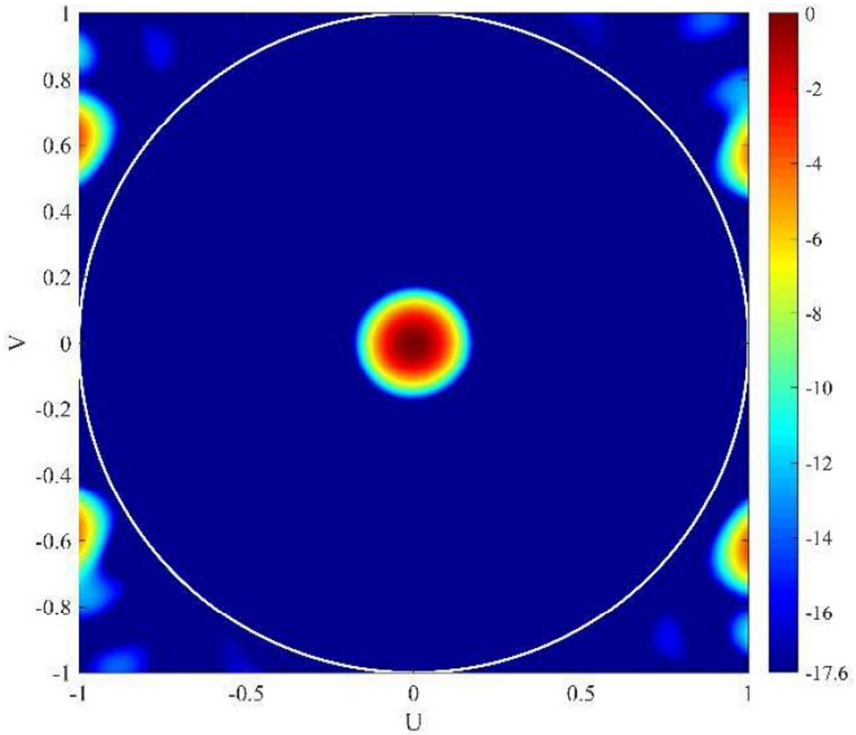

Figure 33. Normalized magnitude in [dB] of the resulting pattern radiated by the array in Figure 31. The visible region is delimited by the withe circle. N_e=35 active radiating elements.

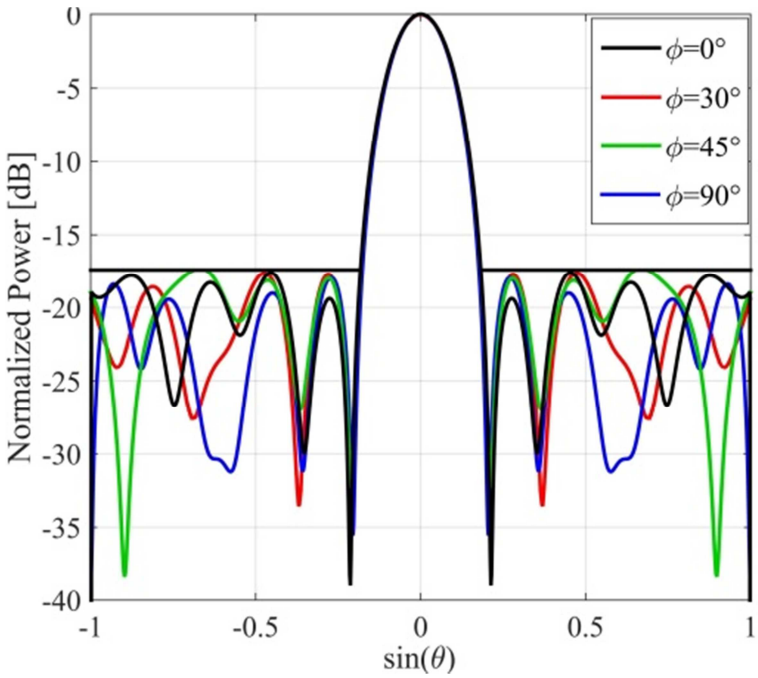

Figure 34. ϕ-cutting planes of the normalized power pattern radiated by the array in Figure 31 with N_e=35 active radiating elements.

No.	(x[λ],y[λ])	w(x,y)	No.	(x[λ],y[λ])	w(x,y)
1	(-2.5, -0.3)	4.84E-01	19	(-0.1, 2.5)	7.79E-01
2	(-2.5, 0.3)	5.69E-01	20	(0.1, 0.0)	8.94E-01
3	(-2.5, 1.8)	3.98E-01	21	(0.2, -1.7)	5.53E-01
4	(-2.2, 1.0)	3.98E-01	22	(0.3, 1.7)	4.57E-01
5	(-2.0, -0.9)	6.28E-01	23	(0.6, -2.5)	5.12E-01
6	(-1.8, 0.0)	7.40E-01	24	(0.7, -0.9)	8.38E-01
7	(-1.7, -1.7)	6.64E-01	25	(0.7, 0.8)	9.88E-01
8	(-1.6, 1.7)	7.55E-01	26	(0.8, 2.5)	6.43E-01
9	(-1.2, -2.5)	6.26E-01	27	(1.1, -1.7)	7.81E-01
10	(-1.2, -0.9)	4.51E-01	28	(1.1, 1.6)	8.98E-01
11	(-1.2, 0.8)	5.59E-01	29	(1.6, -0.8)	5.87E-01
12	(-1.0, 2.5)	4.92E-01	30	(1.7, -2.4)	3.98E-01
13	(-0.8, -1.7)	3.98E-01	31	(2.0, 0.0)	8.23E-01
14	(-0.8, 0.0)	1.00E00	32	(2.0, 1.6)	6.88E-01
15	(-0.7, 1.7)	5.47E-01	33	(2.1, -1.6)	3.98E-01
16	(-0.3, -2.5)	6.43E-01	34	(2.5, -0.8)	5.97E-01
17	(-0.2, -0.8)	6.76E-01	35	(2.5, 0.8)	7.50E-01
18	(-0.1, 0.9)	5.90E-01			

Table 2. Coordinates and normalized weights for the array in Figure 31.

Therefore, the just discussed benchmark comparison has been obtained with a slightly lower SLL and with a reduction of 14.6% in the number of required active radiating elements.

The second benchmark problem concerns the synthesis of a flat-top circular symmetric pattern dated back to [80], where it has been obtained with a square array, composed of 121 half-wavelength spaced elements, whose edge is 5λ. The synthesized array radiates a real and even field obtained by using a Kaiser window. In [46] the same power pattern has been achieved by assuming, in the flat-top region, as reference field to fit the one radiated by the array synthesized in [80], whereas the SLL is below the prescribed level. The resulting sparse array, on a 5λ edge square support, is composed of 100 radiating elements whose minimum interspacing is approximately 0.35λ. Given, by hypothesis, a real and even radiated field, any lower and upper bound constraint on the magnitude of the array factor |AF| is affine and thus convex. Such a condition allows to formulate the problem in the more general way, exploiting a greater number of degrees of freedom to achieve the maximally sparse solution. In particular, it is sufficient to constrain the array factor to satisfy only the prescribed upper and lower bounds both in the flat-top and

the sidelobe region. By applying the proposed synthesis scheme, the prescribed flat-top power pattern has been synthesized by means of only Ne=57 radiating elements whose minimum interspacing and maximum antenna size are, respectively, greater than 0.5λ and 4.24λ. The optimized array layout is shown in Figure 35. In Figure 36 and Figure 37 the 3D and the 2D normalized power pattern, in the (*u*,*v*) domain, are shown respectively, whereas the ϕ-cutting planes for ϕ=0°, 30°, 45°, and 90° are reported in Figure 38.

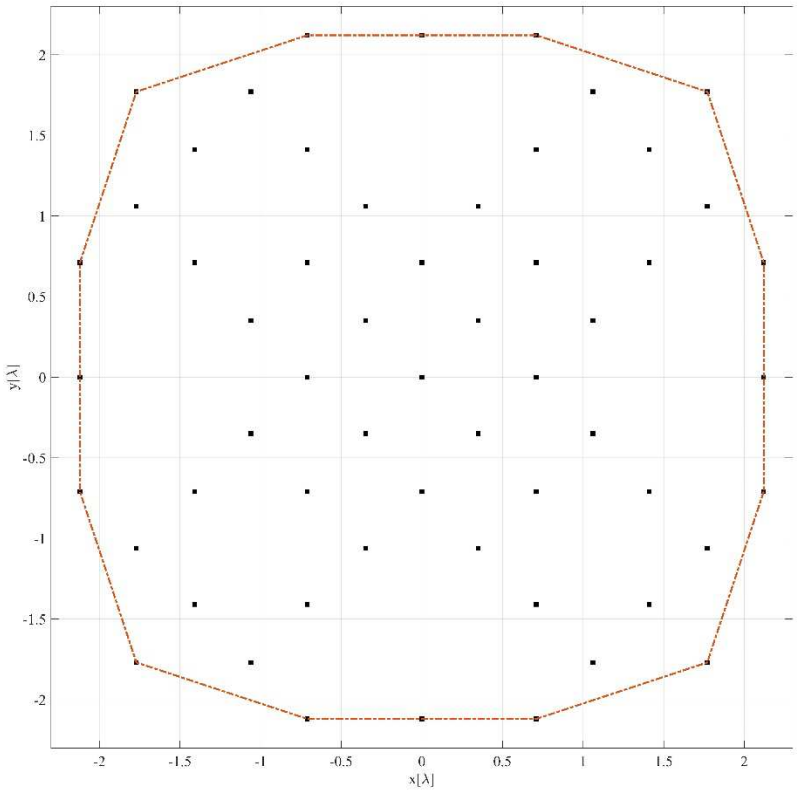

Figure 35. Resulting array layout relative to the synthesis of the flat-top power pattern. Actual radiating elements Ne=57. Discretization grid: dx=dy=0.01λ. The dashed curve represents the convex hull.

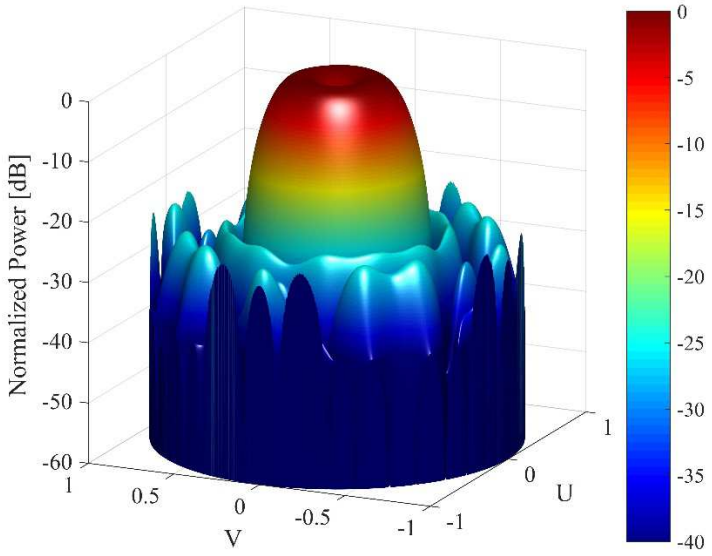

Figure 36. 3D normalized flat-top power pattern radiated by the array in Figure 35 with N_e=57 elements. SLL=-24.53dB.

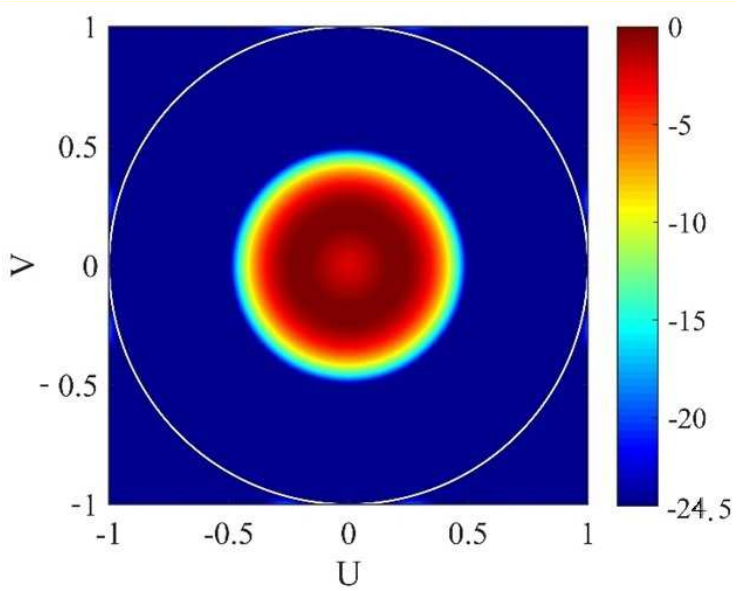

Figure 37. Normalized power in [dB] of the resulting flat-top pattern in the (u,v) domain. The visible region is delimited by the withe circle, Ne=57 active radiating elements.

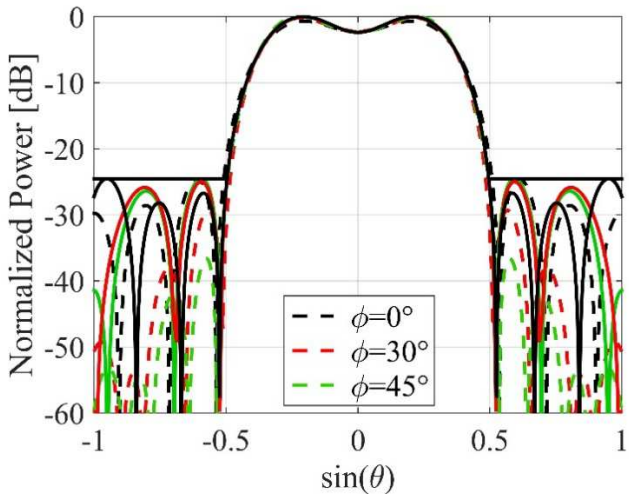

Figure 38. ϕ-cutting planes of the normalized flat-top power pattern synthesized in [80] (dotted lines) and the one synthesized with the proposed synthesis scheme (solid lines). Ne=57 active radiating elements.

Given the symmetry, with respect to the main axes and the bisecting lines of both weights and relative positions of the synthesized array layout, only 11 element positions with the associated weights, whose dynamics is 26.58dB, need to be provided, see Table 3.

No.	(x[λ],y[λ])	w(x,y)	No.	(x[λ],y[λ])	w(x,y)
1	(0.00, 0.00)	1.00E00	7	(1.41, 0.71)	-4.69E-02
2	(0.71, 0.00)	5.95E-01	8	(2.12, 0.71)	-1.25E-01
3	(2.12, 0.00)	-1.69E-01	9	(1.77, 1.06)	-1.01E-01
4	(0.35, 0.35)	5.61E-01	10	(1.41, 1.41)	-7.50E-02
5	(1.06, 0.35)	2.33E-01	11	(1.77, 1.77)	-6.44E-02
6	(0.71, 0.71)	3.92E-01			

Table 3. Coordinates and normalized weights of the flat-top in Figure 35.

Definitely, while keeping the same requirements on the power mask, the proposed synthesis algorithms allow to reduce both the array support's edge (15.2%) and the number of actual radiating elements (43%). Just for completeness sake we are providing the values of Q and the aperture efficiency of the synthesized flat-top pattern, although this is not very common for this type of patterns. In particular, we have Q=1.97dB, and $A_{eff\_flat-top} = 9.92dB - 10\log_{10}(4\pi * 16.0036) = -13.11dB$.

Either the pencil beam and the flat-top patterns have been published on IEEE Transactions on Antennas and Propagation [48]/[R1].
The following benchmark comparison has been published on Progress In Electromagnetics Research Symposium (PIERS) [99]/[R2].
For the following benchmarks we made the hypothesis of conjugate-symmetrical excitations for coherence sake towards the benchmarks chosen for comparison, since all the synthesis made in [81] first, and then in [50], strongly found on this choice. Clearly this is not the only assumption, others exist which do not decrease the number of degrees of freedom a priori as this does. However, our scope for these specific types of patterns was to inspect whether our algorithms could return better performances (in terms of sparsity) than the ones achieved in [50] with as many equal conditions as possible.
Let us consider a rectangular-grid array with a half-wavelength spacing of 14 x 14 elements for the joint synthesis of two shaped patterns, a circular and a diamond-like one, which in [50] have been obtained with 150 radiating elements with SLLs of -25.85dB and -24.3dB respectively. For the first pattern the mainlobe region is specified as a circular-shaped one, with equation $\left\{(u,v):\ u^2+v^2\le 0.2^2\right\}$, whereas the sidelobe one is given by $\left\{(u,v):\ u^2+v^2\ge 0.4^2\right\}$. For the second pattern, which was originally synthesized in [81] with a 14 x 14 array with half-wavelength spacing, the mainlobe region is diamond-shape like, and is defined by $\left\{(u,v):\ |u-0.2|+|v-0.2|\le 0.2\right\}$, whereas the sidelobe one is given by $\left\{(u,v):\ |u-0.2|+|v-0.2|\ge 0.4\right\}$. Furthermore, a circular-shaped null region is required too, and is given by $\left\{(u,v):\ (u+0.5)^2+(v+0.5)^2\le 0.1^2\right\}$, with a null of -50dB. In both cases the required ripple is ≤1dB. All the sidelobe regions are within the visible circle. We would like to remark, in the present discussion, that the aforementioned SLL for the diamond-shaped pattern makes the joint synthesis unfeasible unless the (u,v) spectrum is sampled with 81 x 81 points, as is apparently done in [50]. With the (u,v) spectrum sampled with 128 x 128 points, as we did in our synthesis, the minimum possible SLL is slightly higher, in particular it is -23.67dB. With such sampling and SLL for the diamond-shape pattern, all the constraints are met even when plotting the radiation pattern on more points. With our proposed algorithms we obtained the joint synthesis with only 119 radiating elements, with ENRR=39.29% which is higher than ENRR=23.47% achieved in [50]. Figure 39 and Figure 40 show the layouts obtained in [50] and in this thesis work, respectively. Figure 41, Figure 42 and Figure 43 show the circular 3D/2D patterns and the main axes cuts respectively, whereas Figure 44, Figure 45 and Figure 46 show the diamond-shaped 3D/2D patterns and the main axes cuts respectively. In particular, in Figure 46 the cuts are apparently centered at $\theta=asin(0.2)$. For the circular pattern the Q factor is approximately 5dB, whereas the directivity of the synthesized array is of 14.19dB. Therefore we have $A_{eff\_circ\_pattern}$=14.19dB-10log_{10}(4π*42.25)=-13.05dB. For the diamond-shaped one we have Q=4.57dB and $A_{eff\_diamond\_pattern}$=9.56dB-10log_{10}(4π*42.25)=-17.69dB.

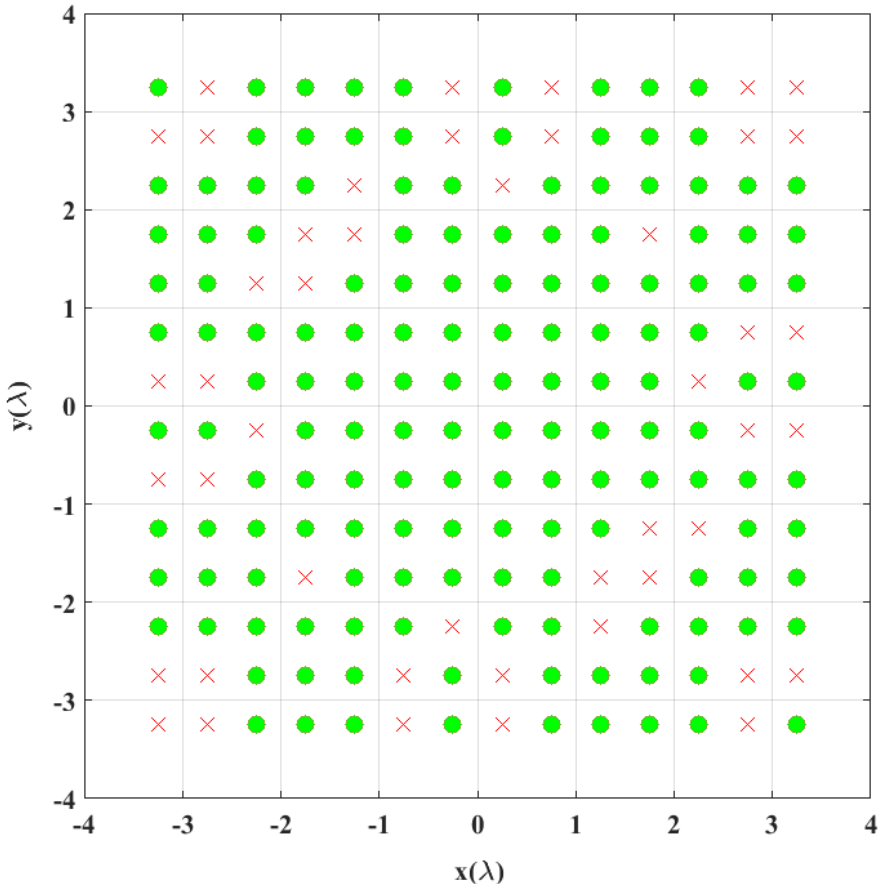

Figure 39. Array of 150 elements obtained in [50]

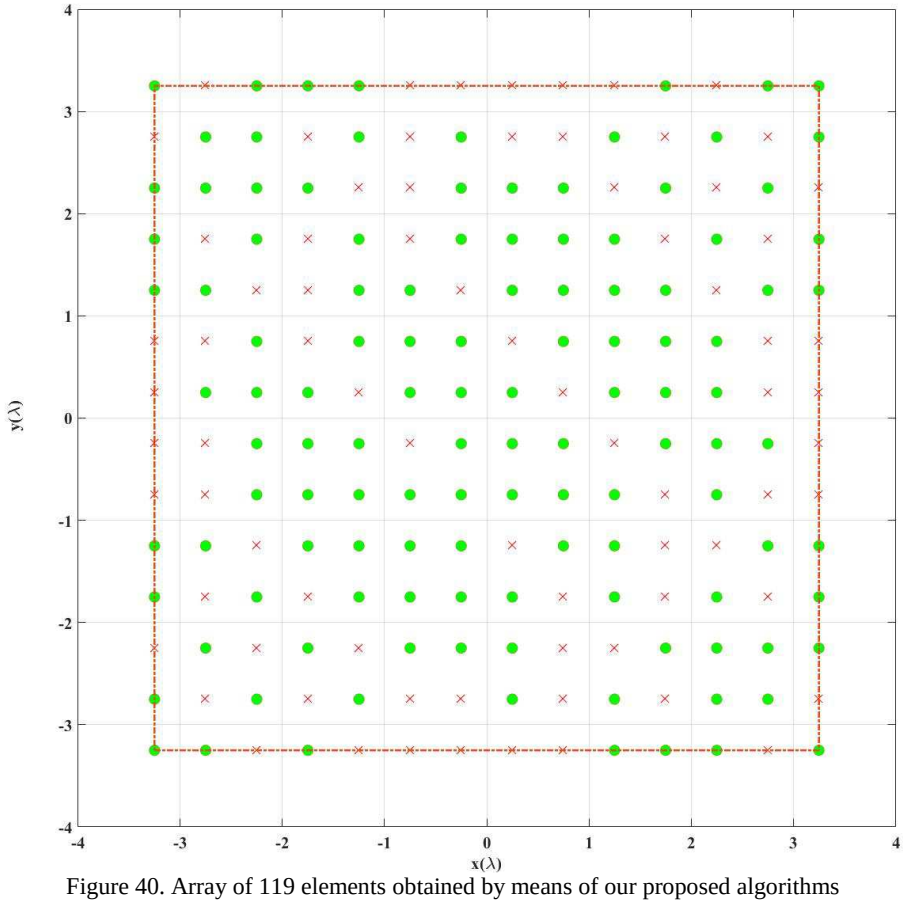

Figure 40. Array of 119 elements obtained by means of our proposed algorithms

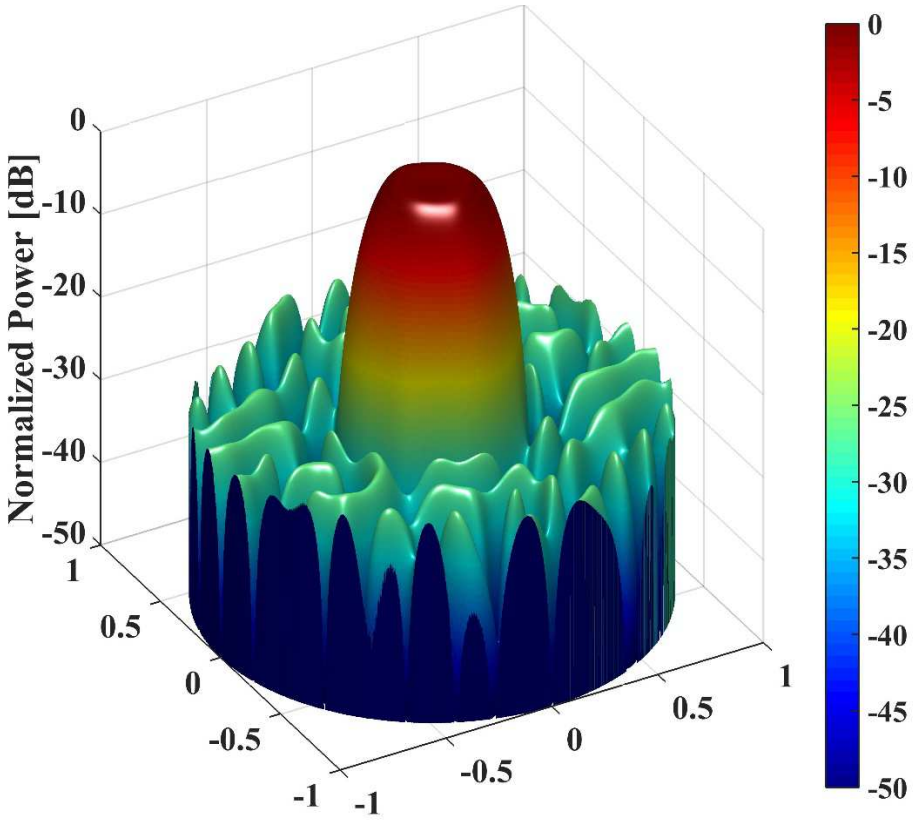

Figure 41. Circular shape 3D pattern

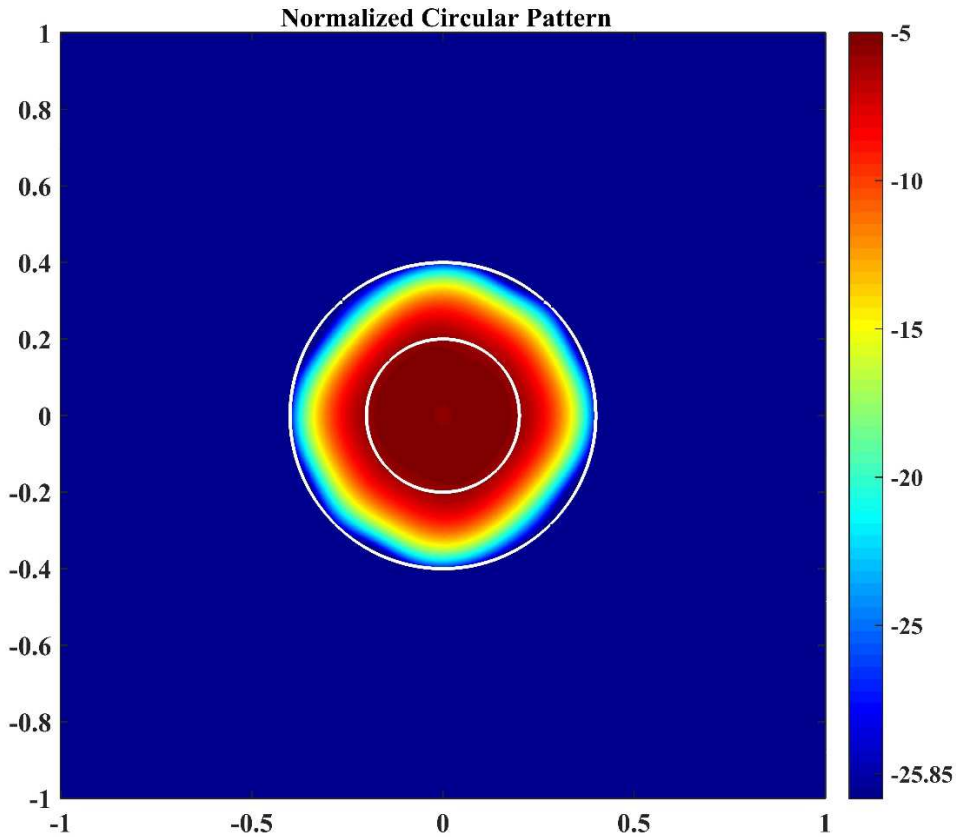

Figure 42. 2D circular pattern. The two white circles represent the mainlobe and SL boundaries respectively

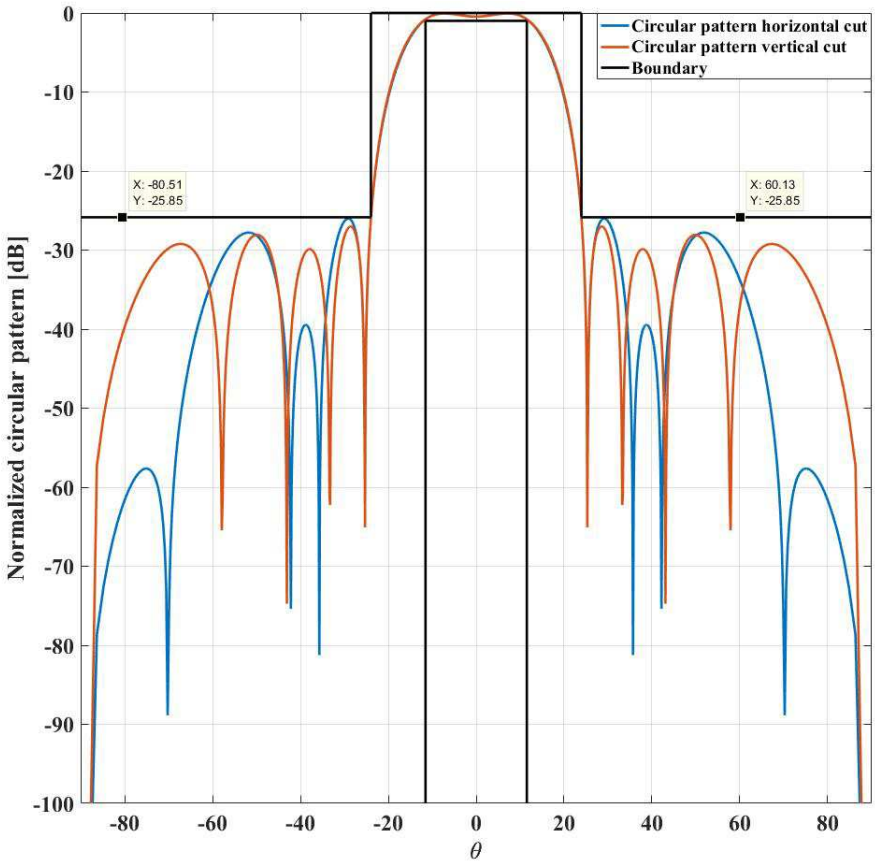

Figure 43. 2D circular pattern cuts. The vertical (u = 0) and horizontal (v = 0) cuts are reported in the figure together with the boundary required by the benchmark. Datatips have been inserted to show precisely the SLL.

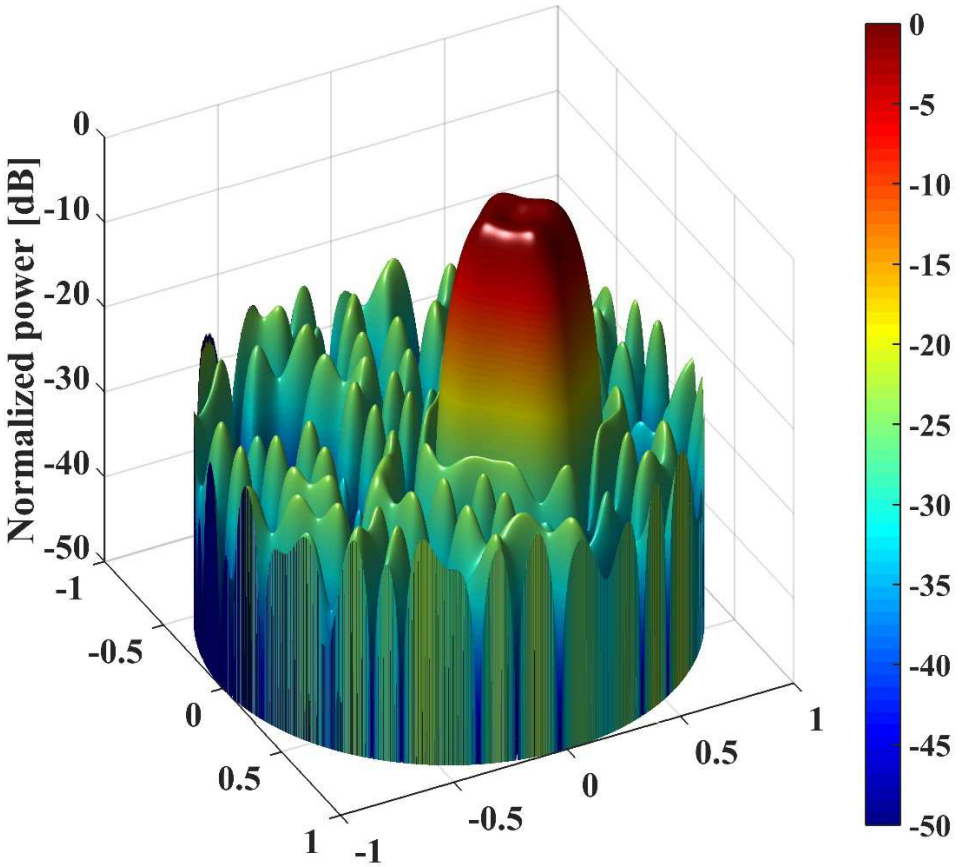

Figure 44. Diamond-shape 3D pattern

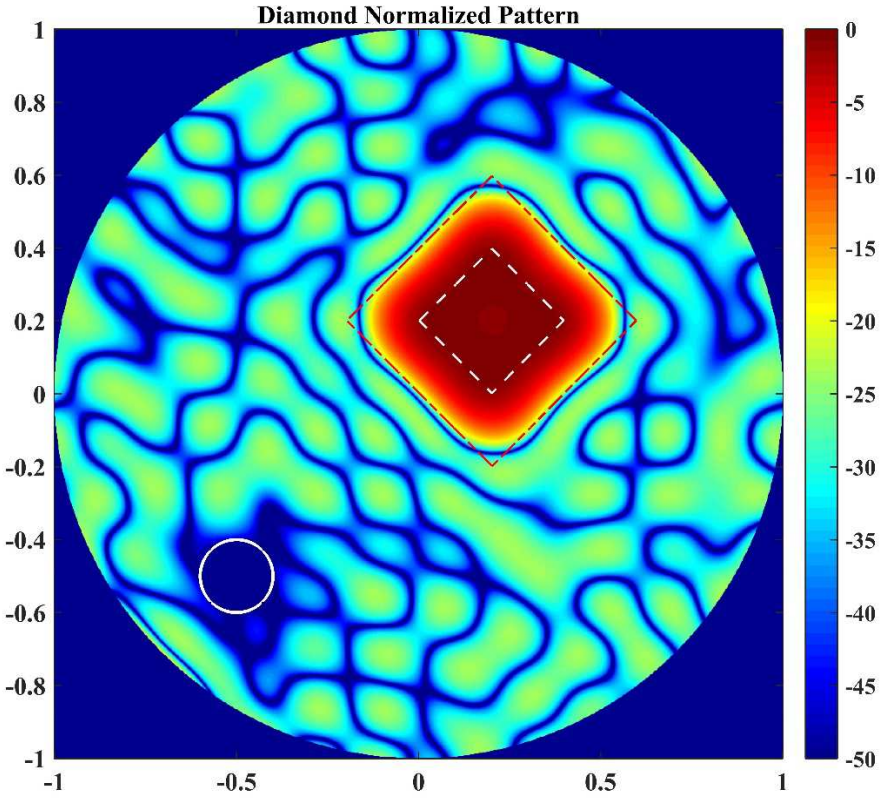

Figure 45. 2D diamond-shape pattern. The white circle and dashed square, together with the red dash-dotted square represent the null region, the mainlobe and sidelobe regions boundaries respectively.

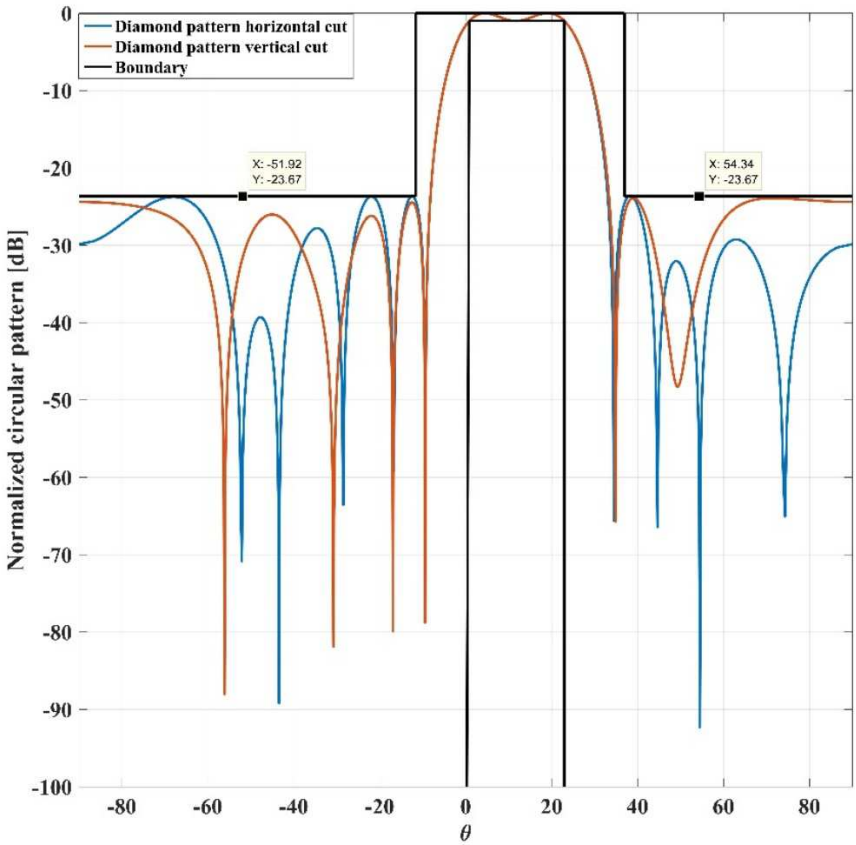

Figure 46. 2D diamond-shape cuts. The vertical (u = 0) and horizontal (v = 0) cuts are reported in the figure together with the boundary required by the benchmark. Datatips have been inserted to show precisely the SLL.

4.3 New benchmarks

In this section we are assessing the performances of the proposed algorithms for a series of cases that involve the variations of the antenna size and steering range in the visible circle after fixing its geometry, BW and optimum SLL (the latter as described in section 3.2). The BWs are referred to the highest SLL level and reported in degrees as double-sided, whereas the maximum steering directions are determined according to the main lobe extension in order to avoid the rise of grating lobes. The full array is composed of elements whose distance is half-wavelength, and acts as reference for the computation of the ENRR. In Table 4 the input parameters are grouped.

Size	BW [deg]	SLL [dB]	F
3λ	39.8643	-22.9784	49
5λ	25.3326	-24.3877	121
7λ	18.5734	-25.0483	225

Table 4. Input parameters: array edge, BWs and SLLs

The results of the performed numerical analysis are synthesized in Figure 47, whose steering range varies in [0°,57.06°]. The trend shown in Figure 47 seems to confirm that the more degrees of freedom the algorithms can sparsify on, which corresponds to larger and larger antenna sizes, the higher is the reduction of the elements, therefore representing a major advantage for electrically large arrays.
Coherently with the examples discussed in the previous section, the Q factor of the synthesized arrays is always higher than the one pertaining to the same array in the isophoric case, thus exhibiting a smaller robustness in terms of weight dynamics. Moreover, the directivity of all the synthesized, sparse arrays is in any case smaller than the one of their isophoric counterpart (as expected).

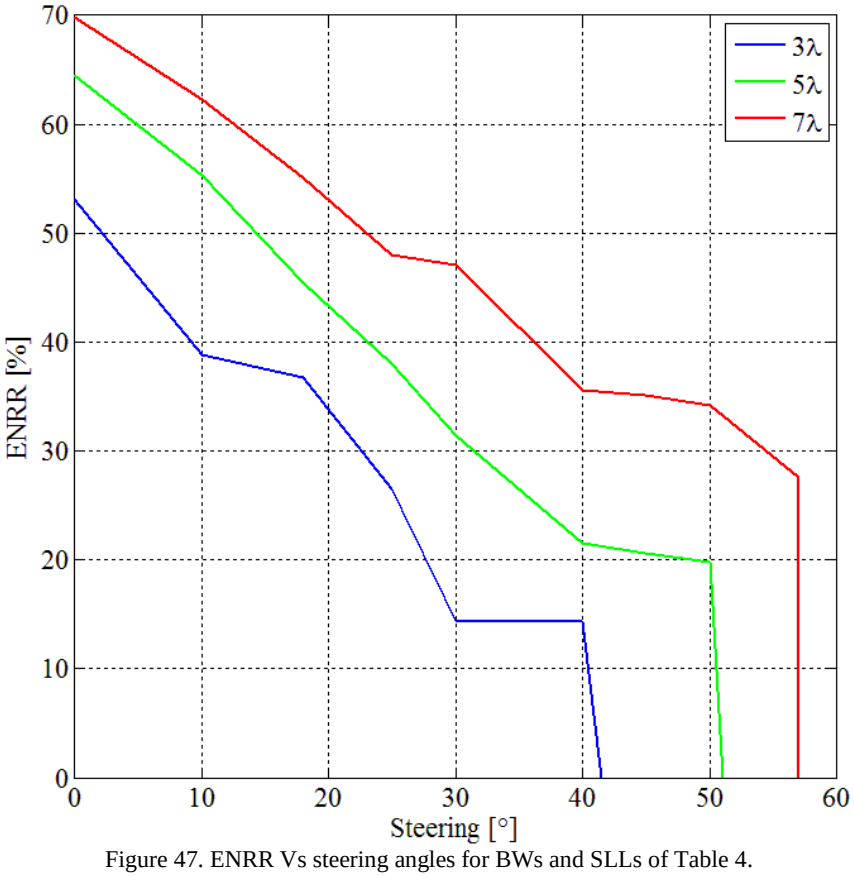

Figure 47. ENRR Vs steering angles for BWs and SLLs of Table 4.

In Figure 47 when the ENRR converges to zero means that the full array solution is achieved, which corresponds to the highest steering ranges, which are, for the 3λ, 5λ and 7λ arrays, 41.5°, 51° and 57.06° respectively. Please note that the last different from zero values of ENRR refer to steering angles that are

very close to the above maximum ones; in the 7λ case for instance the maximum steering angle relative to the last non-zero value of the ENRR is 57°, hence the two values are barely distinguishable. Clearly, since only the latter case can reach 57° of steering, it represents the most complete case, and thus has been chosen for a more detailed description. Figure 48 shows the array layout which is characterized of N_e=163 and a relative ENRR of 27.56%, whereas Figure 49 shows the 3D normalized power pattern lying in the region $\Phi = \left\{(u,v): u^2+v^2 \le \left[1+\sin(57°)\right]^2\right\}$ when u_0=v_0=0. The ϕ-cutting planes for ϕ=0°, 30°, 45° and 90° are reported in Figure 50. Since the upper bound in Table 4 has been imposed in the bi-dimensional region Φ, the relative synthesized array is able to radiate a pencil pattern up to 57° along any direction, see Figure 17. Figure 51 shows the normalized 2D power pattern when the main beam is steered along (u_0,v_0)=[sin(57°),0]. As it can be seen, this steering direction is such that no grating lobes enter the visible region and, in particular, that it is the maximum one allowed by the array under discussion.
The weight dynamics is 7.21dB, the Q factor and the directivity of the sparse synthesized array are, respectively, 1.36dB and 23.1dB, whereas the ones in the isophoric case 1.36dB and 23.4dB (thus confirming, as expected, the synthesized array is non-superdirective). The area bounded by the convex hull is 43.5λ^2, which causes an aperture efficiency loss of 23.1dB-24.377dB=-1.277dB.

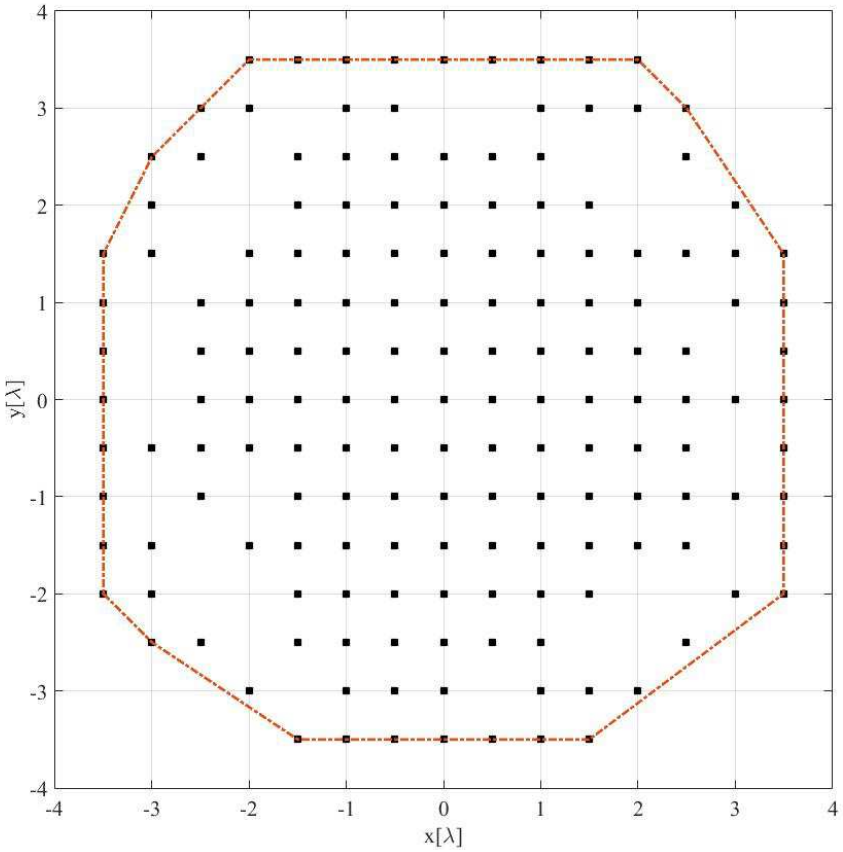

Figure 48. Resulting steerable array relative to the synthesis of pencil beam power pattern. Active radiating elements N_e=163, edge size 7λ, steering range 57°. The dash-dotted curve represents its convex hull

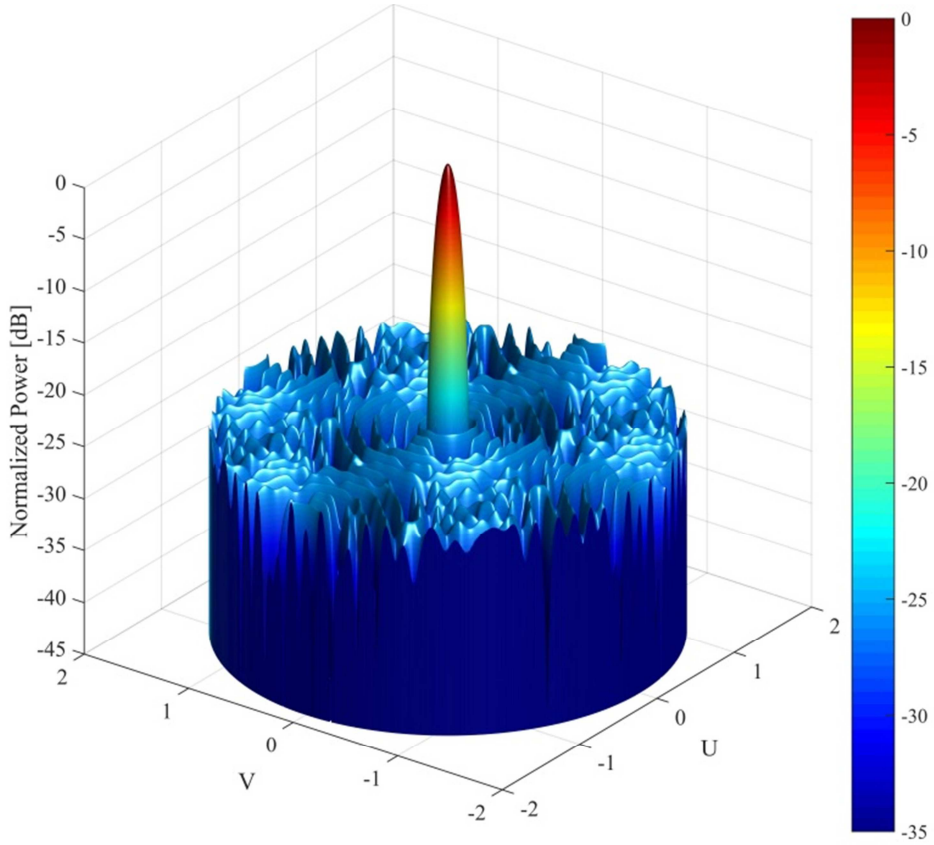

Figure 49. 3D power pattern radiated by the 7λ steerable array (Figure 48) pointing at broadside. Ne=163 active radiating elements. BW=18.57°, SLL≅-25dB

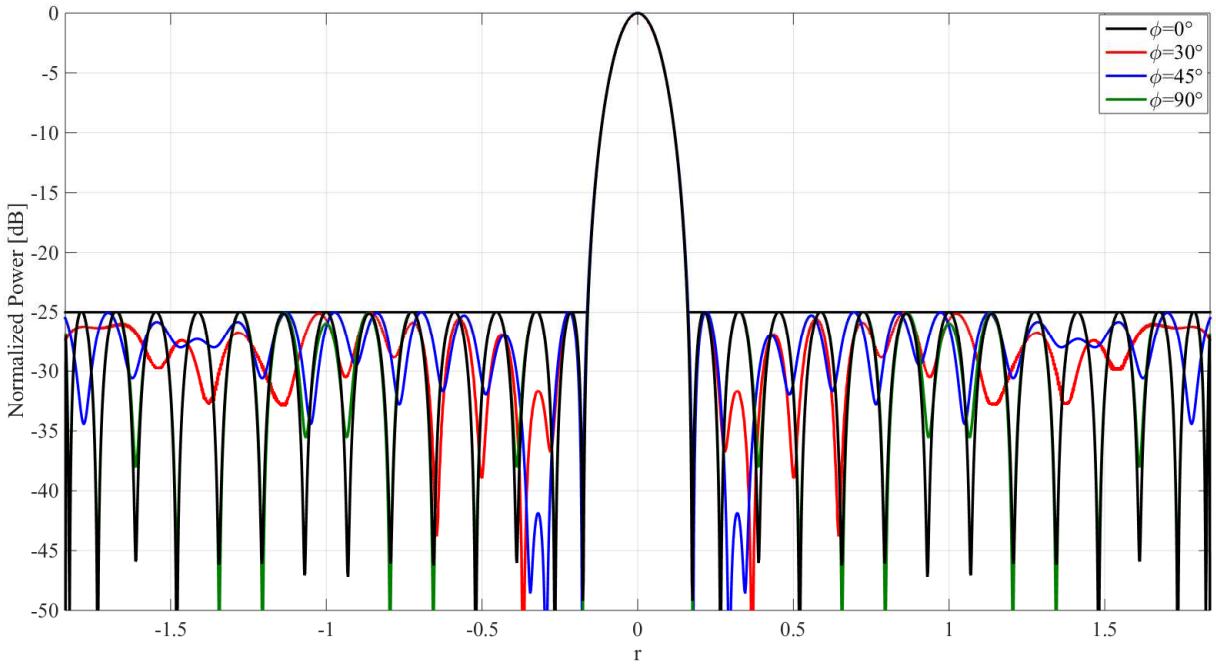

Figure 50. ϕ-cutting planes for ϕ=0°, 30°, 45° and 90° of the normalized power pattern of Figure 49. On the abscissa it is $r \in [-1-\sin(57°), 1+\sin(57°)]$

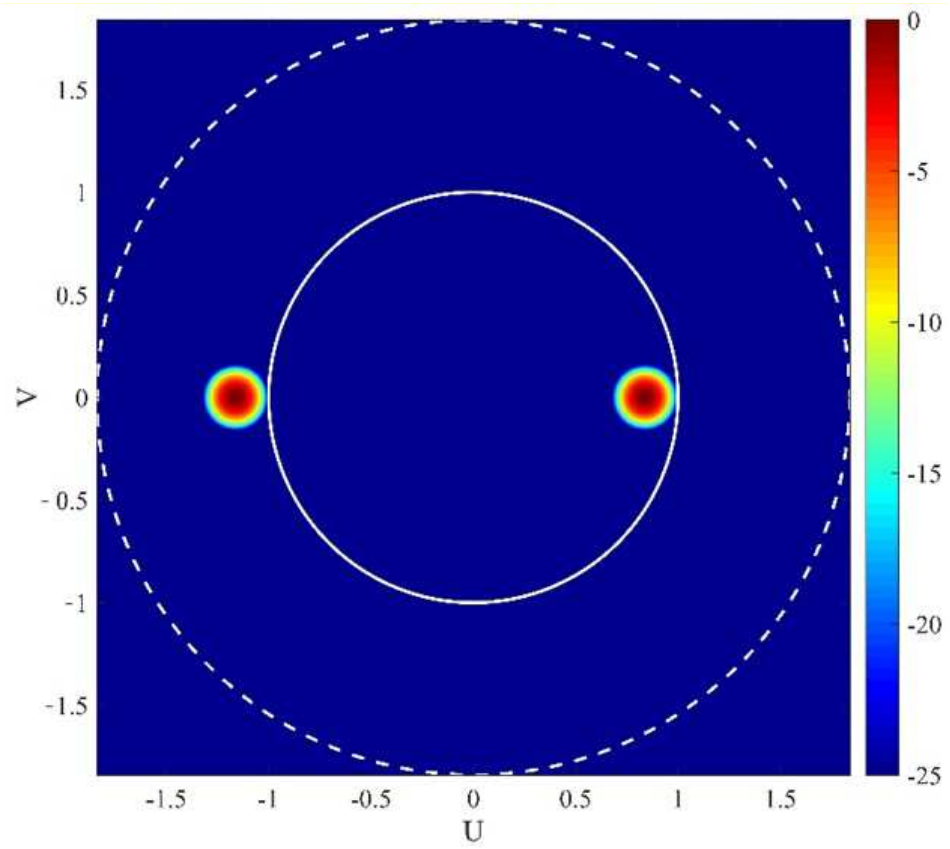

Figure 51. 2D power pattern of the 7λ steerable array depicted in Figure 48 pointing at $(u_0,v_0)=[sin(57°),0]$ and with the input parameters in Table 4. The visible region is delimited by the white circle, whereas the dashed one identifies the region Φ

Chapter 5

Experimental assessment of the algorithms application: Far Field measurements in anechoic chamber

5.1 Introduction

In this chapter we are dealing with the experimental validation of the algorithms described in the previous ones by means of measurements in anechoic chamber of the chosen antenna, that for secrecy reasons we are naming as *X-antenna*. The sparse synthesis has been carried out by switching off the elements of the standard, full antenna, thus it falls within the special case of thinning. Clearly indeed it would not have been possible to perform measurements of a sparse version of the *X-Antenna* with the positions of the elements that could fall out of the existing ones. A picture of the latter is given in Figure 52, (sources at [101]). First we are showing how the array simulated by a full-wave simulator as Ansys HFSS meets the requirements and compares to the predicted performances obtained in Matlab, then we are comparing the simulation results to the measurements. Either the synthesis and the HFSS simulations have been carried out by taking into account the element factor of the array. The latter has been simulated individually in HFSS (with master/slave conditions) and all the necessary electric field components have been exported to Matlab for the synthesis process, which obviously has been achieved jointly in Tx and Rx.

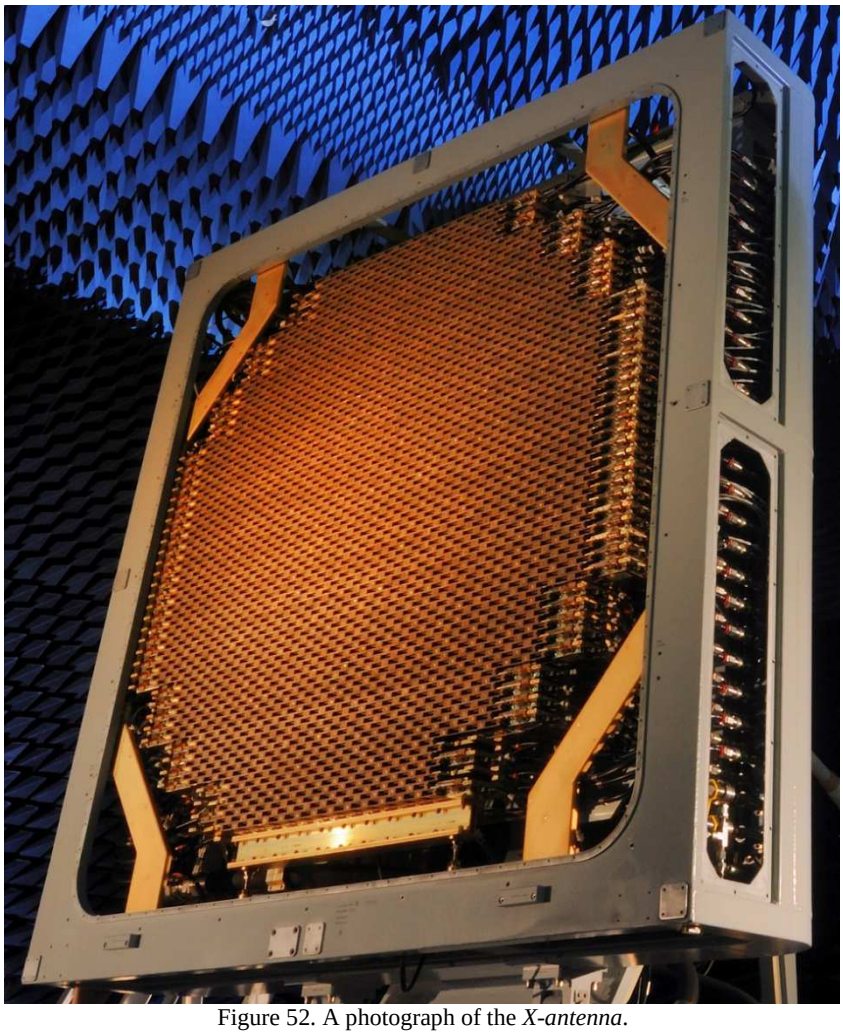

Figure 52. A photograph of the *X-antenna*.

5.2 The X-antenna: features and requirements

The *X-antenna* operates in C-band, is an active full phased array with Tx/Rx solid state modules, and is able to steer up to ±45° in Azimuth and ±60° in elevation. Among the different beams it can form we chose the pencil-type sum one for this thesis work. For secrecy reasons all the requirements that follow are parametrized, as well as other quantities like:

- the frequency at which the synthesis has been made, which is f_c;
- the number of elements of the full *X-antenna*, which we name as $N_{X\text{-}full}$.

5.2.1 The Tx requirements

All the following requirements are referred to the maximum pattern value.

- Beamwidth @-3dB for horizontal and vertical cuts: LFBT
- Beamwidth @-3dB for ±45° Azimuth (v=0 cut): LFAT
- Beamwidth @-3dB for ±60° Elevation (u=0 cut): LFET
- SLL within ±LTF for horizontal and vertical cuts: ≤LLBT
- SLL beyond ± LTF for horizontal and vertical cuts (in at least 90% of the whole observation interval): ≤LLFBT
- rms SLL beyond ±LTF for horizontal and vertical cuts: ≤LLRBT
- SLL within ±LT for ±45° Azimuth (v=0 cut): ≤LLAT
- SLL beyond ±LT for ±45° Azimuth (v=0 cut) (in at least 90% of the whole observation interval): ≤LLFAT
- rms SLL beyond ±LT for ±45° Azimuth (v=0 cut): ≤LLRAT
- SLL within ±LT for ±60° Elevation (u=0 cut): ≤LLET
- SLL beyond ±LT for ±60° Elevation (u=0 cut) (in at least 90% of the whole observation interval): ≤LLFET
- rms SLL beyond ±LT for ±60° Elevation (u=0 cut): ≤LLRET

5.2.2 The Rx requirements

All the following requirements are referred to the maximum pattern value.

- Beamwidth @-3dB for horizontal and vertical cuts: LFB
- Beamwidth @-3dB for ±45° Azimuth (v=0 cut): LFA
- Beamwidth @-3dB for ±60° Elevation (u=0 cut): LFE
- SLL within ±LTF for horizontal and vertical cuts: ≤LLB
- SLL beyond ±LTF for horizontal and vertical cuts (in at least 90% of the whole observation interval): ≤LLFB
- rms SLL beyond ±LTF for horizontal and vertical cuts: ≤LLRB
- SLL within ±LT for ±45° Azimuth (v=0 cut): ≤LLA
- SLL beyond ±LT for ±45° Azimuth (v=0 cut) (in at least 90% of the whole observation interval): ≤LLFA
- rms SLL beyond ±LT for ±45° Azimuth (v=0 cut): ≤LLRA
- SLL within ±LT for ±60° Elevation (u=0 cut): ≤LLE
- SLL beyond ±LT for ±60° Elevation (u=0 cut) (in at least 90% of the whole observation interval): ≤LLFE
- rms SLL beyond ±LT for ±60° Elevation (u=0 cut): ≤LLRE
- Tapering dynamics ≤TR

5.3 Comparison between simulated results: the Tx Synthesis

In Figure 53 the full array in Tx is depicted (only two colors), whereas in Figure 54 the sparse synthesized array is shown. It shall be considered indeed that since in anechoic chamber the measurements could not be carried out by physically removing the radiating elements that were not supposed to be present (that is, the “switched-off” ones with 0 weight), the HFSS simulations have been carried out by setting 0 power in place of the absent elements, so that these are simply passive (only physically present). The ENRR is 55.74%, and most of the requirements listed in 5.2.1 have been met. In particular:

- the beamwidth @-3dB for the horizontal cut is LFBT+0.0014° in HFSS, thus it is slightly wider than the expected value. Anyway, the requirement is not met.
- The beamwidth @-3dB for the vertical cut is LFBT-0.0451° in HFSS.
- The beamwidth @-3dB for ±45° Azimuth (v=0 cut) is LFAT+0.2228° in HFSS, thus the requirement is not met.
- The beamwidth @-3dB for ±60° Elevation (u=0 cut) is LFET+0.1775° in HFSS, thus the requirement is not met.
- The rms SLL beyond ±LTF for the horizontal cut is LLRBT-8.38dB
- The rms SLL beyond ±LTF for the vertical cut is LLRBT-3.89dB
- The rms SLL beyond ±LT for 45° Azimuth (v=0 cut) is LLRAT-2.59dB
- The rms SLL beyond ±LT for -45° Azimuth (v=0 cut) is LLRAT-2.58dB
- The rms SLL beyond ±LT for 60° Elevation (u=0 cut) is LLRET-2.01dB
- The rms SLL beyond ±LT for -60° Elevation (u=0 cut) is LLRET-2dB

All the other requirements listed in 5.2.1 are shown in figures from Figure 55 to Figure 60. In Figure 59 and Figure 60 the mask has not been plotted for secrecy reasons.

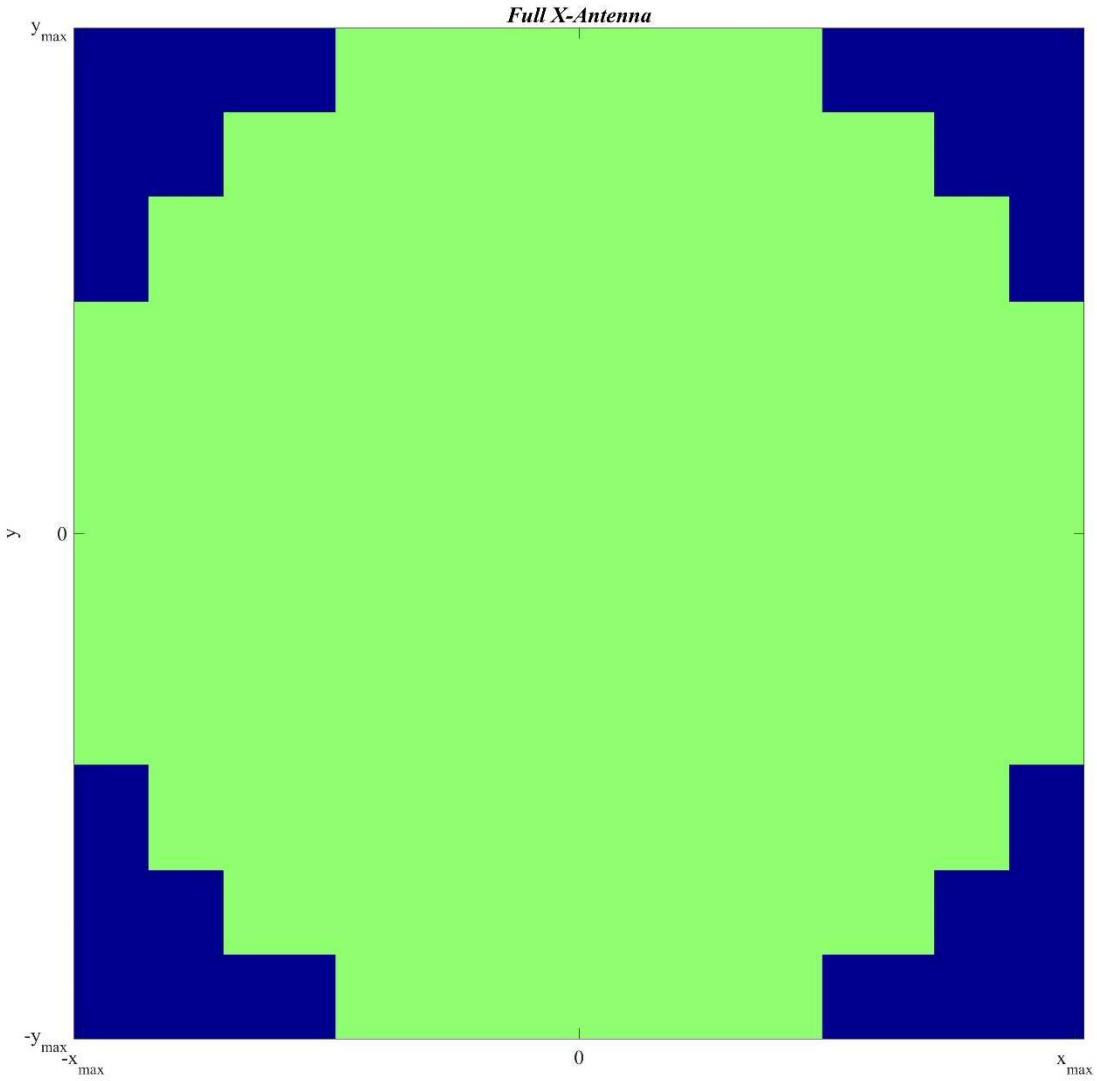

Figure 53. The full *X-Antenna* Tx layout

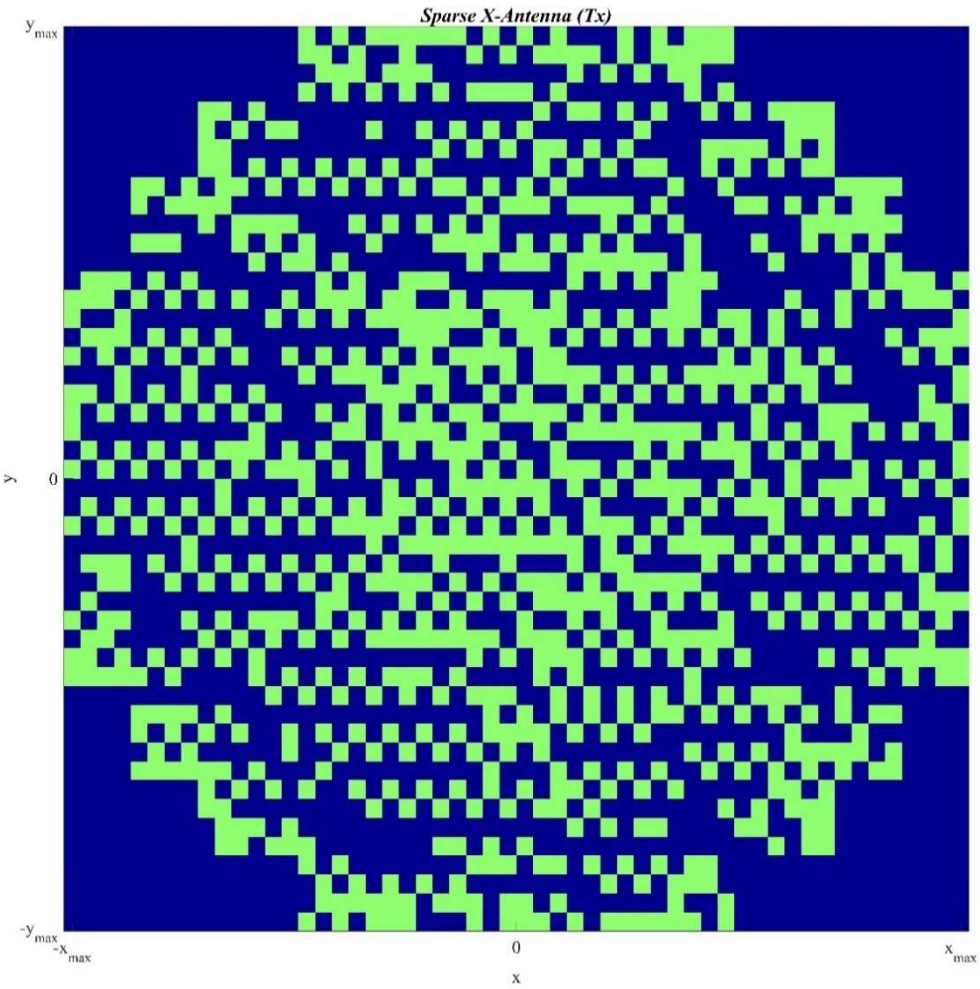

Figure 54. The sparse *X-Antenna* Tx layout

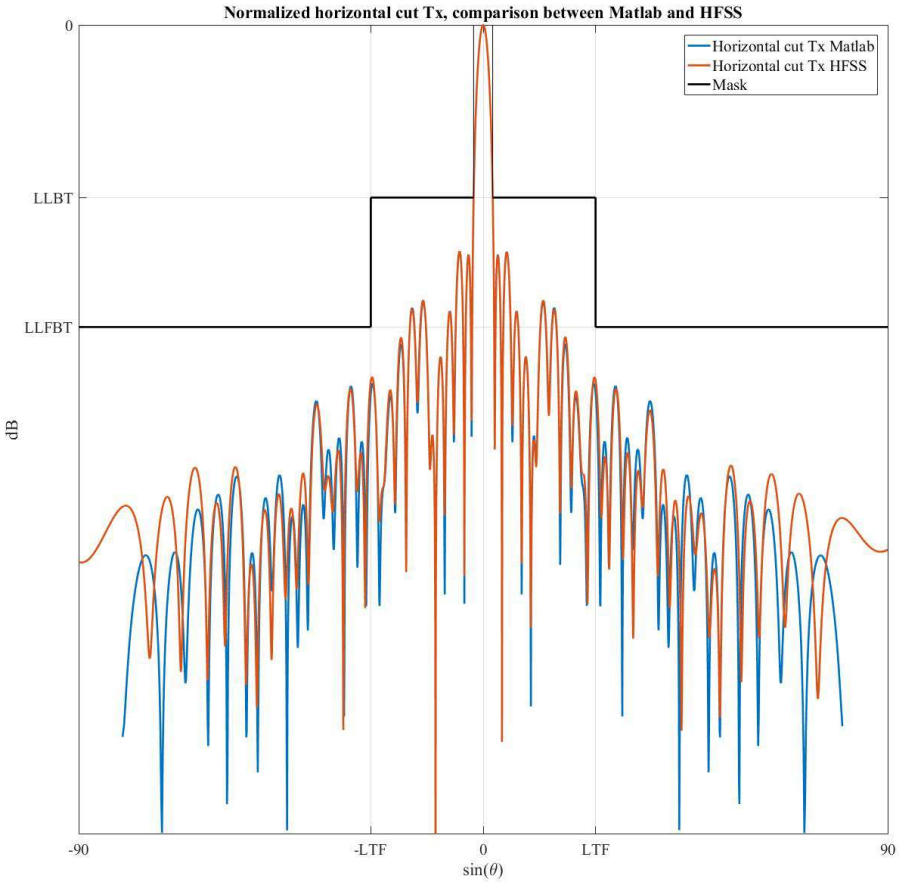

Figure 55. Horizontal cut for sparse Tx *X-Antenna* at broadside

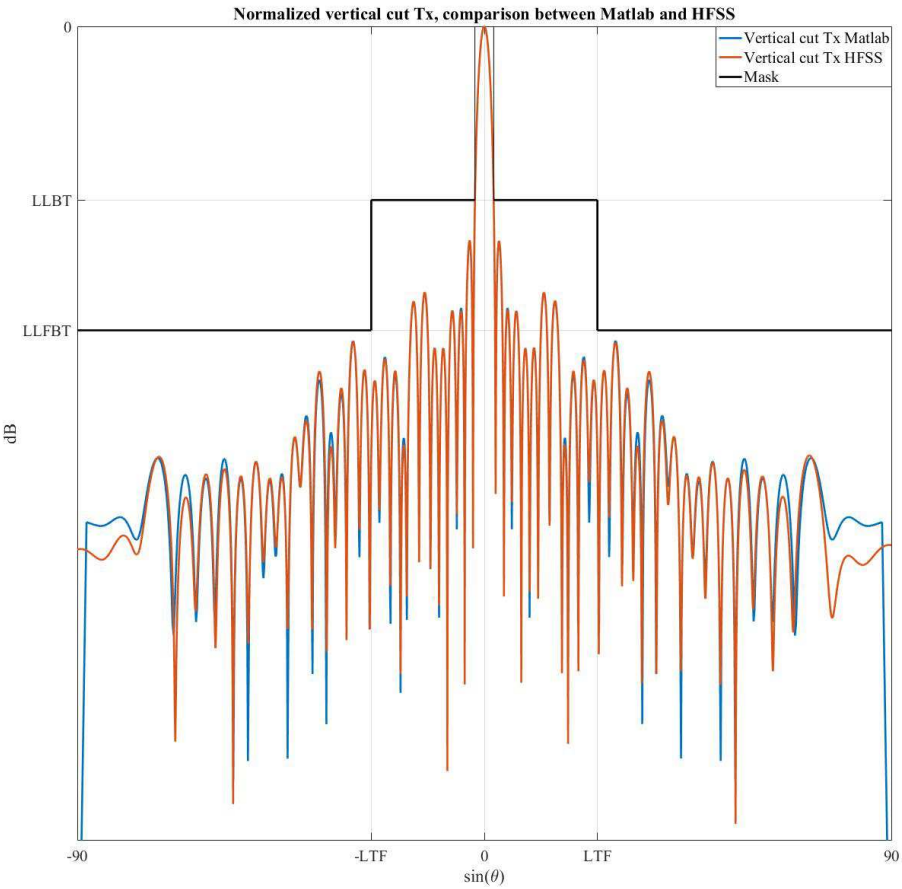

Figure 56. Vertical cut for sparse Tx *X-Antenna* at broadside.

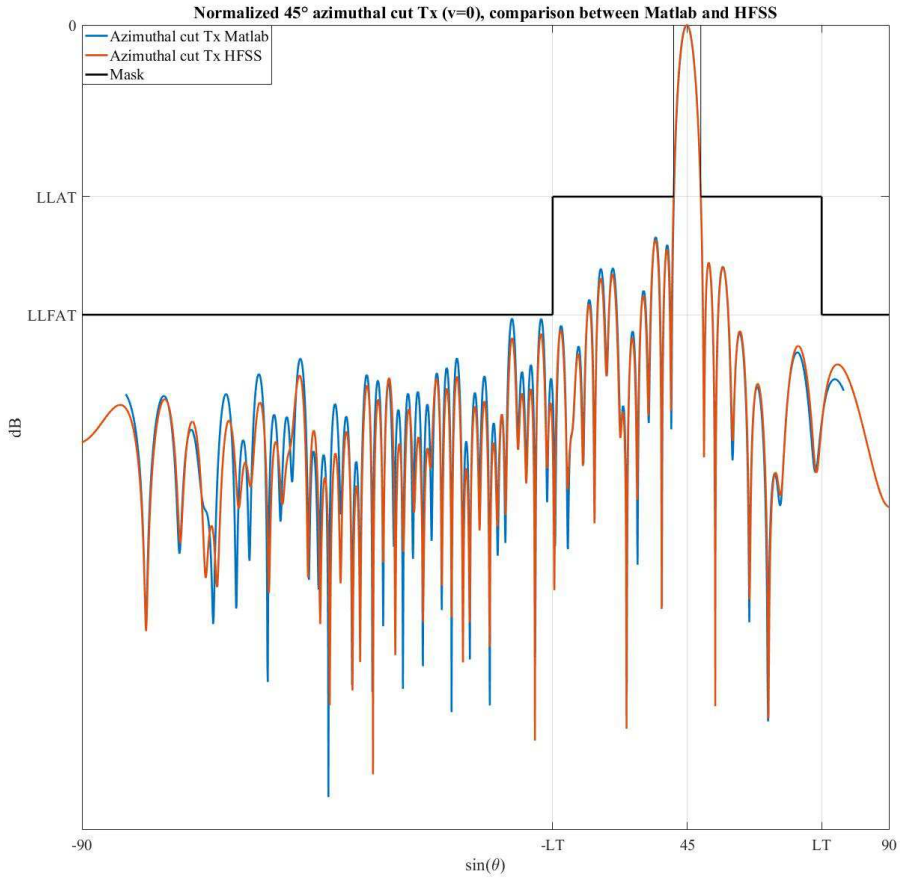

Figure 57. 45° azimuthal (v=0) cut for sparse Tx *X-Antenna*.

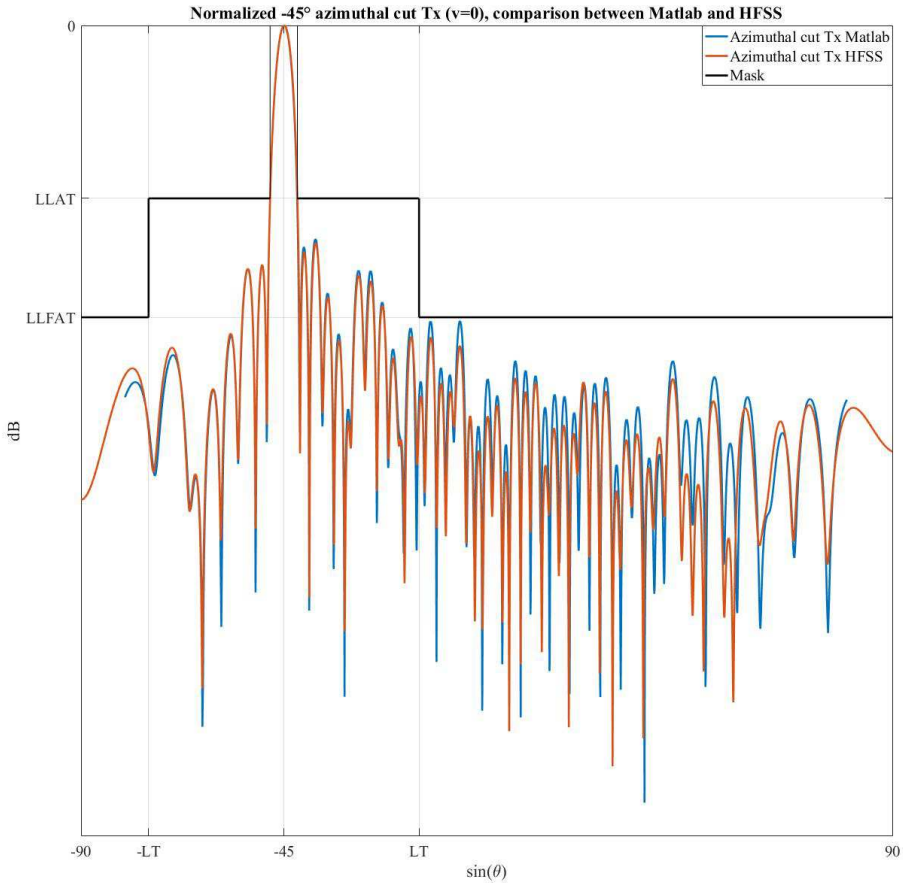

Figure 58. -45° azimuthal (v=0) cut for sparse Tx *X-Antenna*.

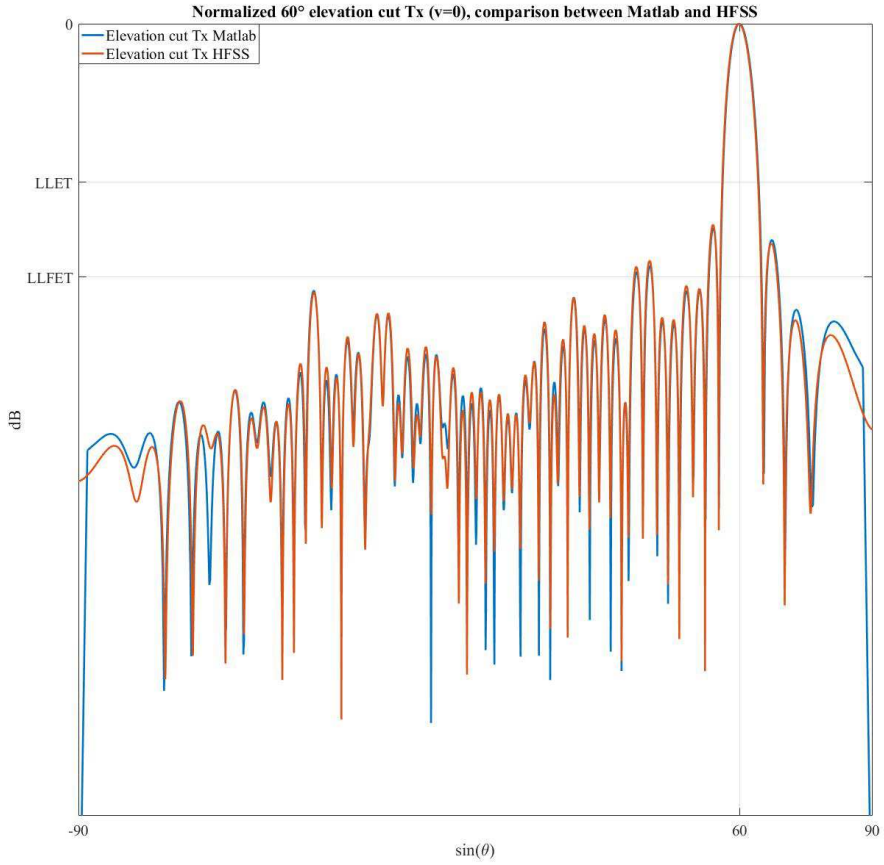

Figure 59. 60° elevation (u=0) cut for sparse Tx *X-Antenna*.

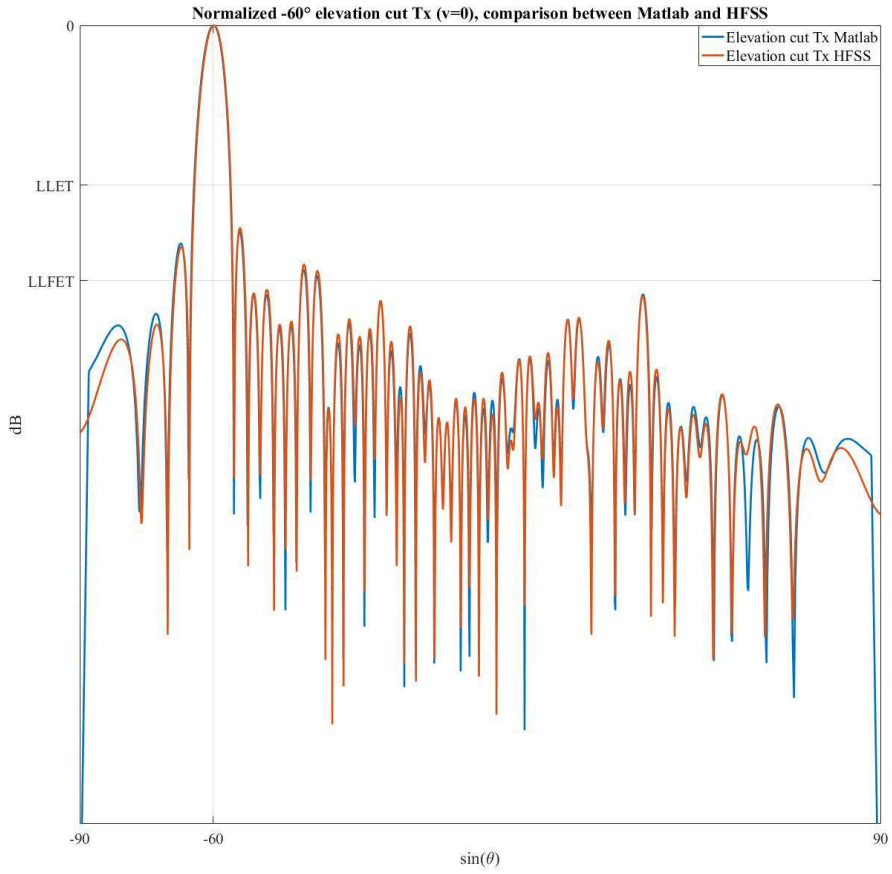

Figure 60. -60° elevation (u=0) cut for sparse Tx *X-Antenna*.

5.4 Comparison between simulated results: the Rx Synthesis

In particular:

- the beamwidth @-3dB for the horizontal cut is LFBT-0.1964° in HFSS.
- The beamwidth @-3dB for the vertical cut is LFB-0.246° in HFSS.
- The beamwidth @-3dB for ±45° Azimuth (v=0 cut) is LFAT-0.2196° in HFSS.
- The beamwidth @-3dB for ±60° Elevation (u=0 cut) is LFET+0.3668° in HFSS.
- The rms SLL beyond ±LTF for the horizontal cut is LLRB-17.57dB
- The rms SLL beyond ±LTF for the vertical cut is LLRB-12.96dB
- The rms SLL beyond ±LT for 45° Azimuth (v=0 cut) is LLRA-13.34dB
- The rms SLL beyond ±LT for -45° Azimuth (v=0 cut) is LLRA-13.32dB
- The rms SLL beyond ±LT for 60° Elevation (u=0 cut) is LLRE-13.5dB
- The rms SLL beyond ±LT for -60° Elevation (u=0 cut) is LLRE-13.49dB
- The tapering dynamics is TR-8.07dB.

Figure 61 shows the tapering of X-Antenna, whereas figures from Figure 62 to Figure 67 show the cuts at broadside, horizontal and vertical, the cuts at ±45° azimuth and ±60° elevation. The last two figures (±60° elevation) do not show the mask for secrecy reasons.

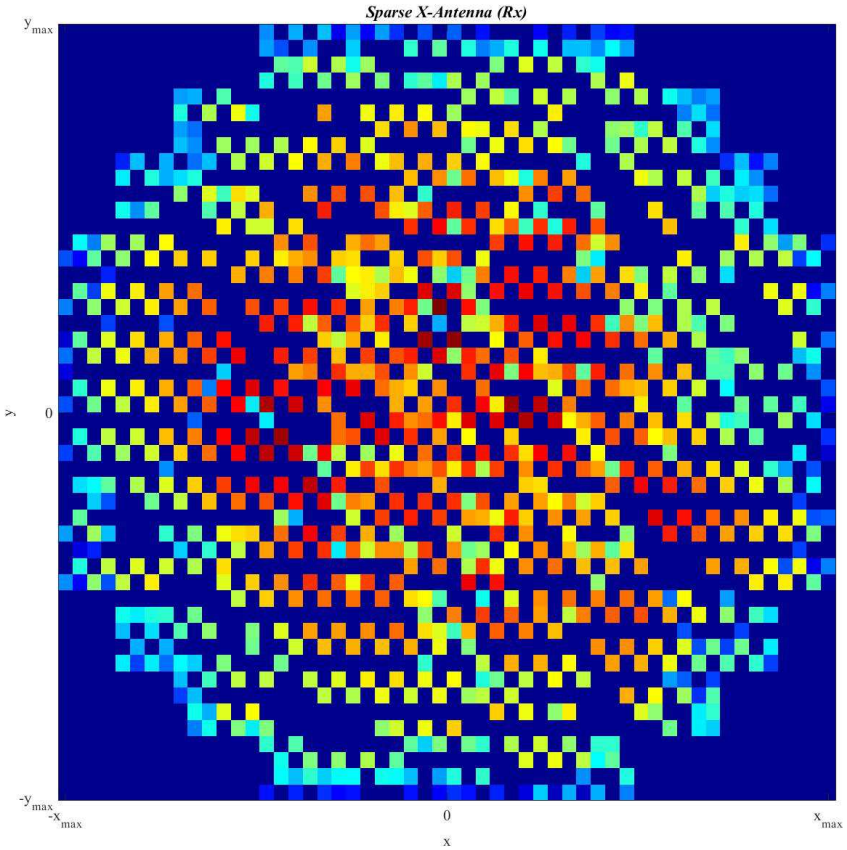

Figure 61. The sparse *X-Antenna* Rx layout

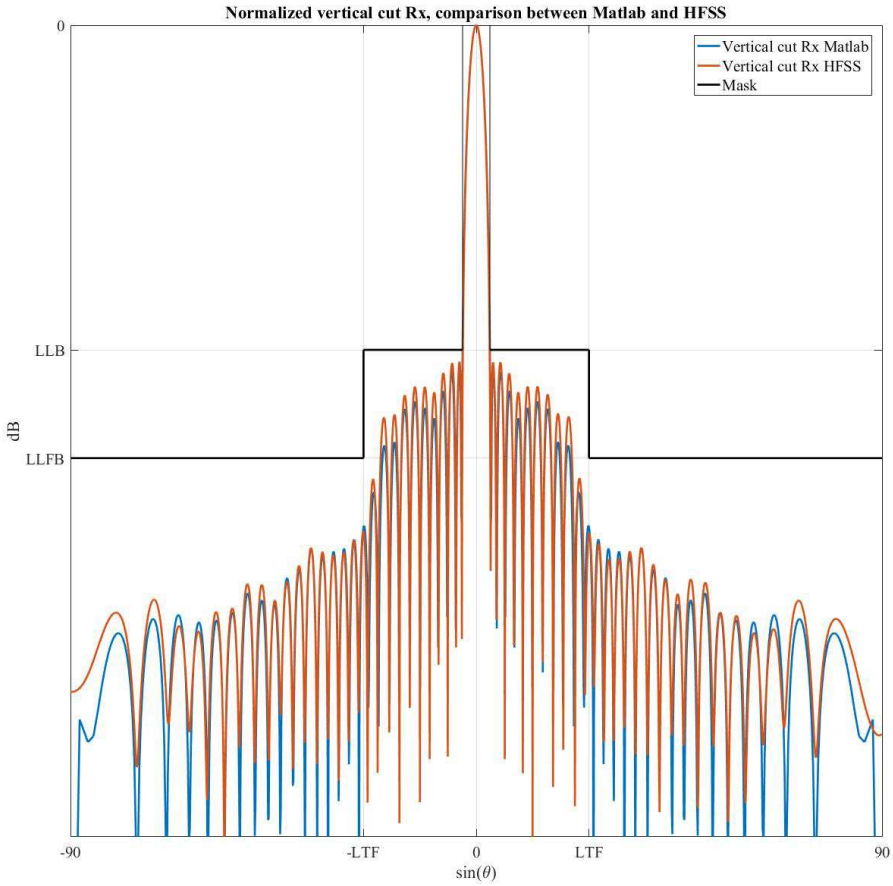

Figure 62. Vertical cut for sparse Rx *X-Antenna* at broadside

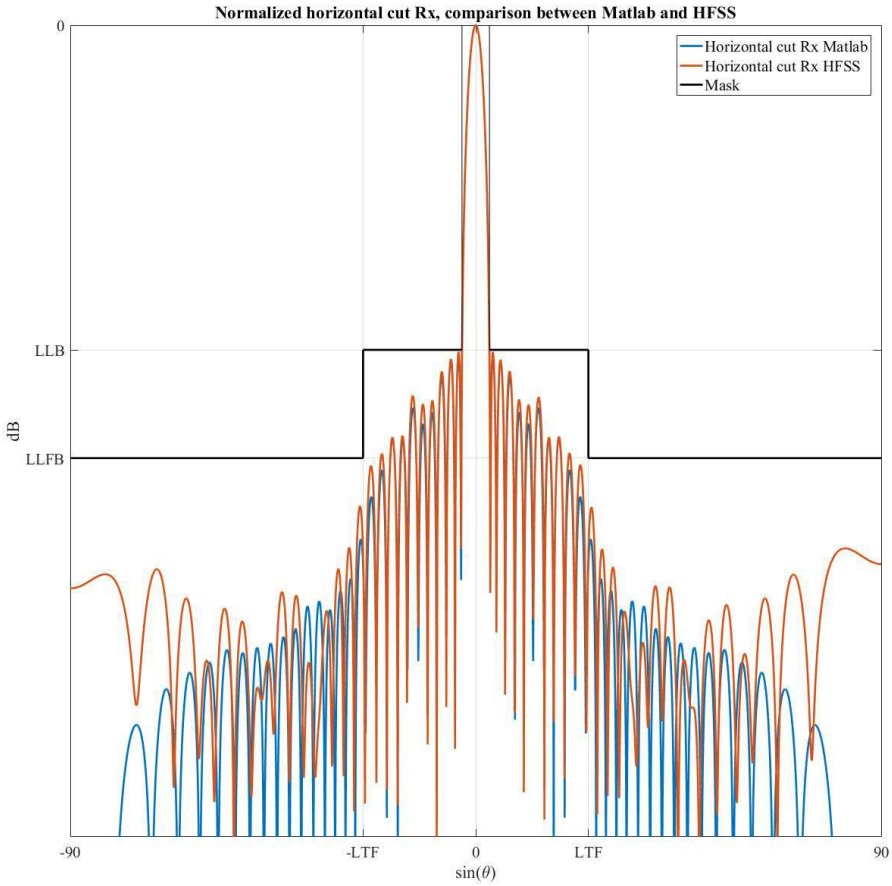

Figure 63. Horizontal cut for sparse Rx X-Antenna at broadside.

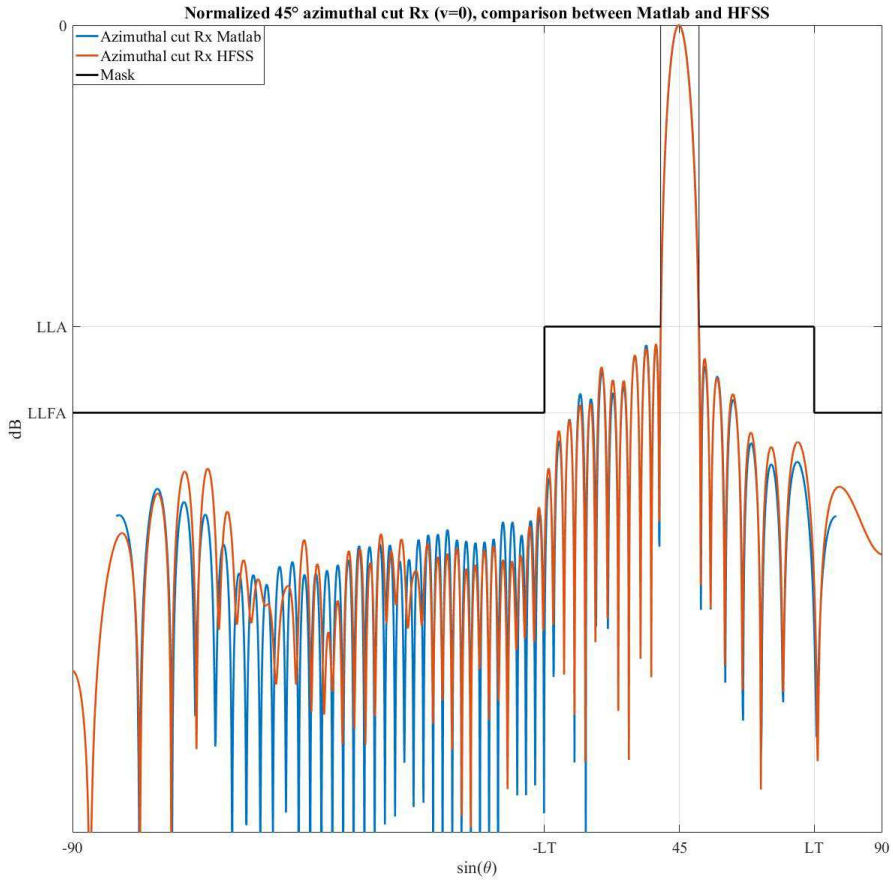

Figure 64. 45° azimuthal (v=0) cut for sparse Rx *X-Antenna*.

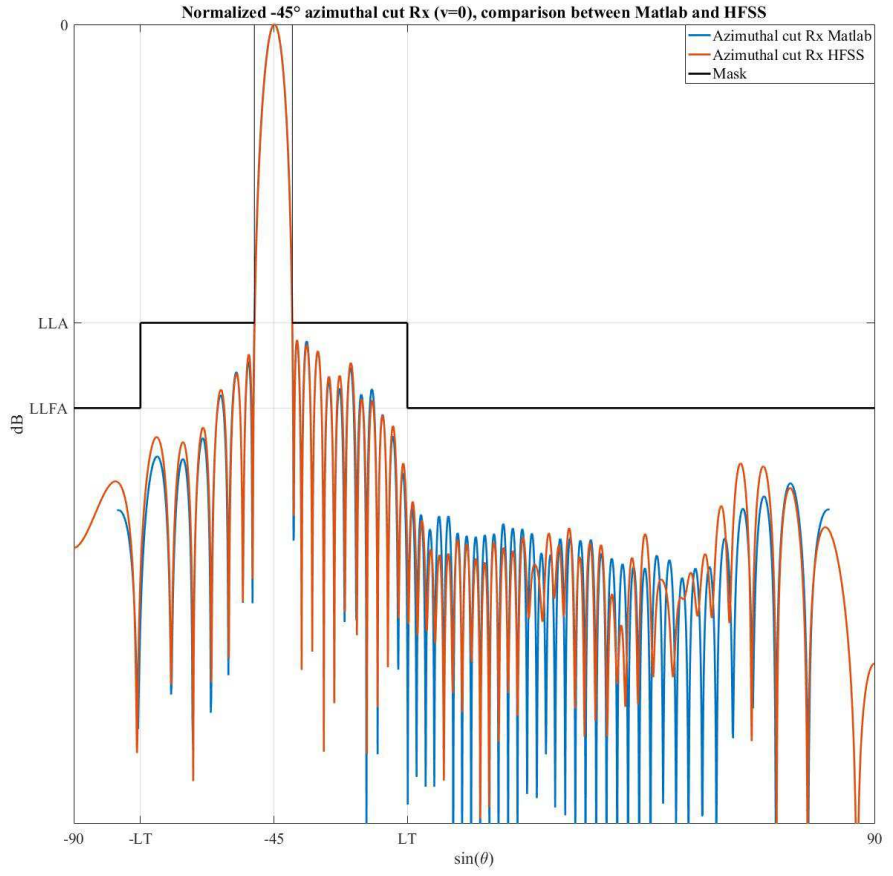

Figure 65. -45° azimuthal (v=0) cut for sparse Rx X-Antenna.

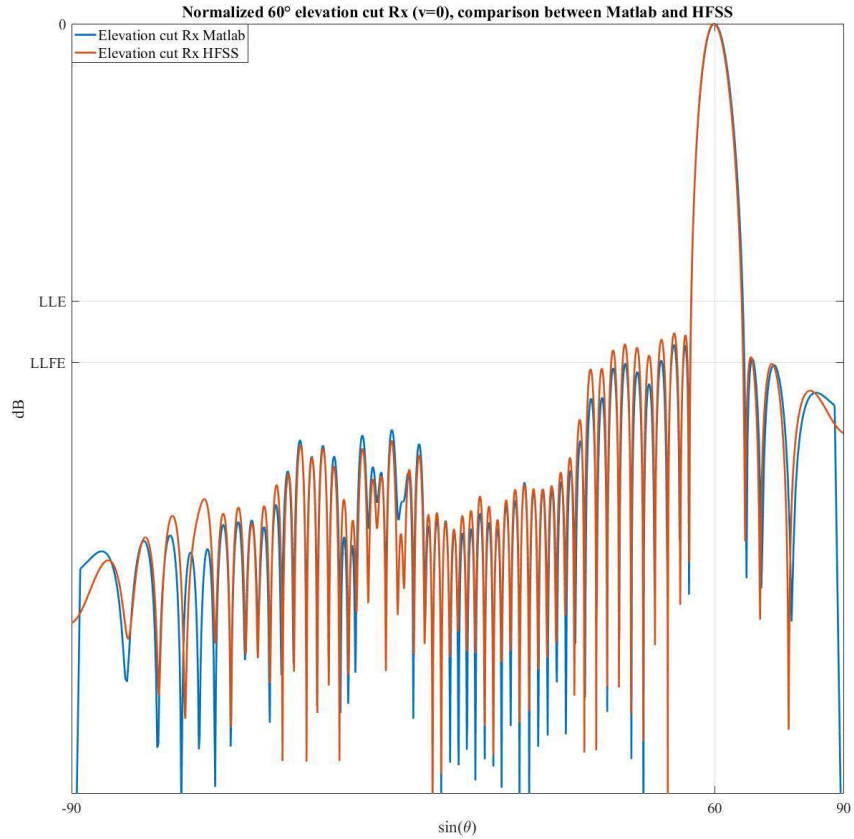

Figure 66. 60° elevation (u=0) cut for sparse Rx *X-Antenna*.

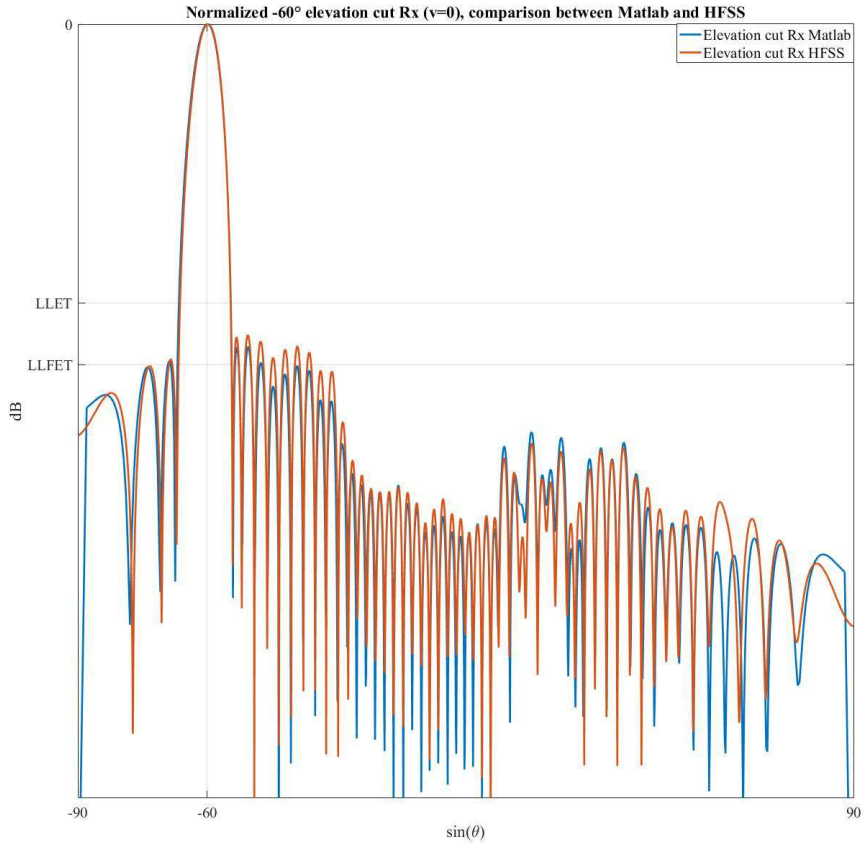

Figure 67. -60° elevation (u=0) cut for sparse Rx *X-Antenna*

5.5 Comparison between simulated and measured results

In this section a comparison between measured and simulated FF is shown at broadside and for one case of steering in order to validate the simulation results. In particular, the plots shown refer to the sparse case in Tx, which is given a more in-depth description in 5.6. For the HFSS curves the simulations have been done by using the actual phases used for making the antenna corrections aimed at approximating, as much as possible, a set of phases of reference for standard directions of steering (like the ones involved in the present discussion). In fact, apart from the theoretical phases (which in the case of broadside can be 0° for each active element for instance) the real ones for the *X-antenna* are typically different from 0, even in the broadside case. The overall comparison shows a quite good coherence between the two types of results.

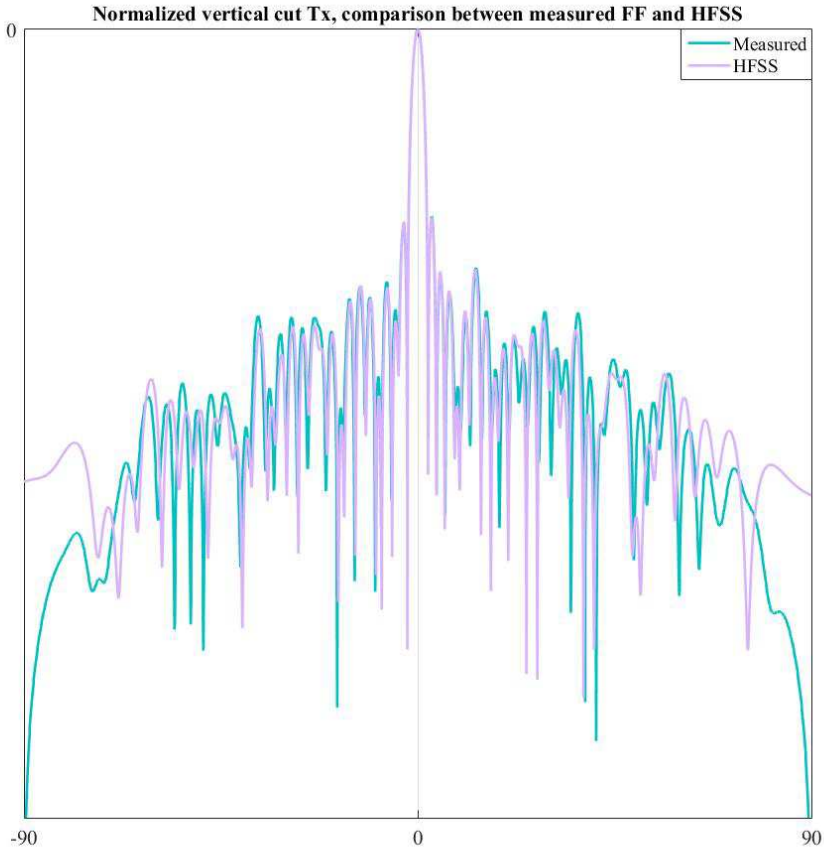

Figure 68. Vertical cut for sparse Tx *X-Antenna* at broadside: comparison between measured and simulated (HFSS) results

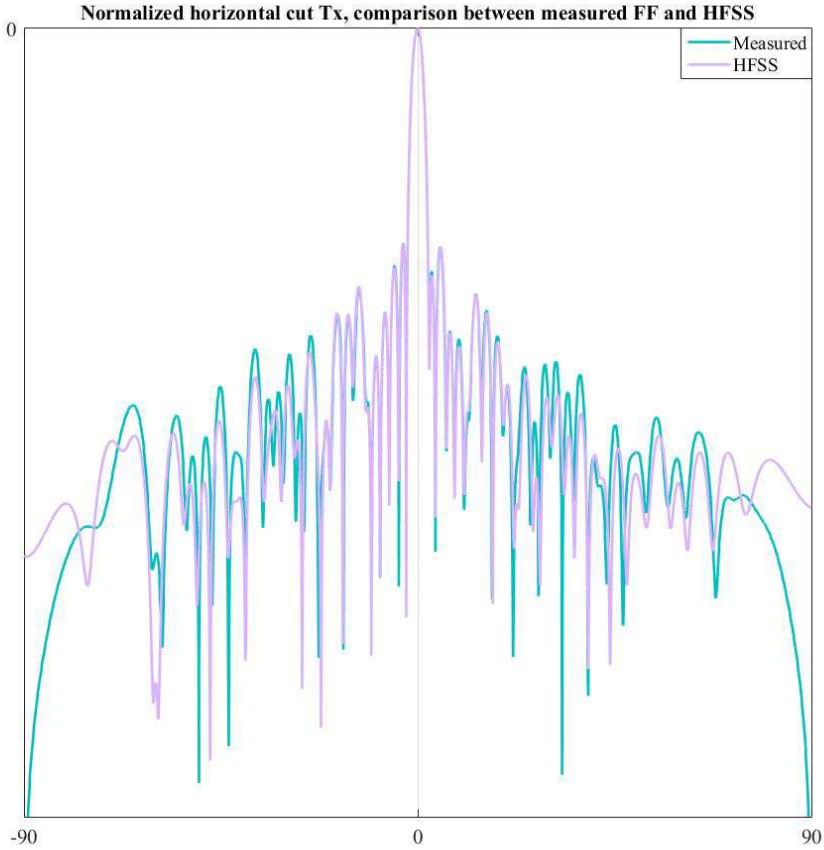

Figure 69. Horizontal cut for sparse Tx *X-Antenna* at broadside: comparison between measured and simulated (HFSS) results

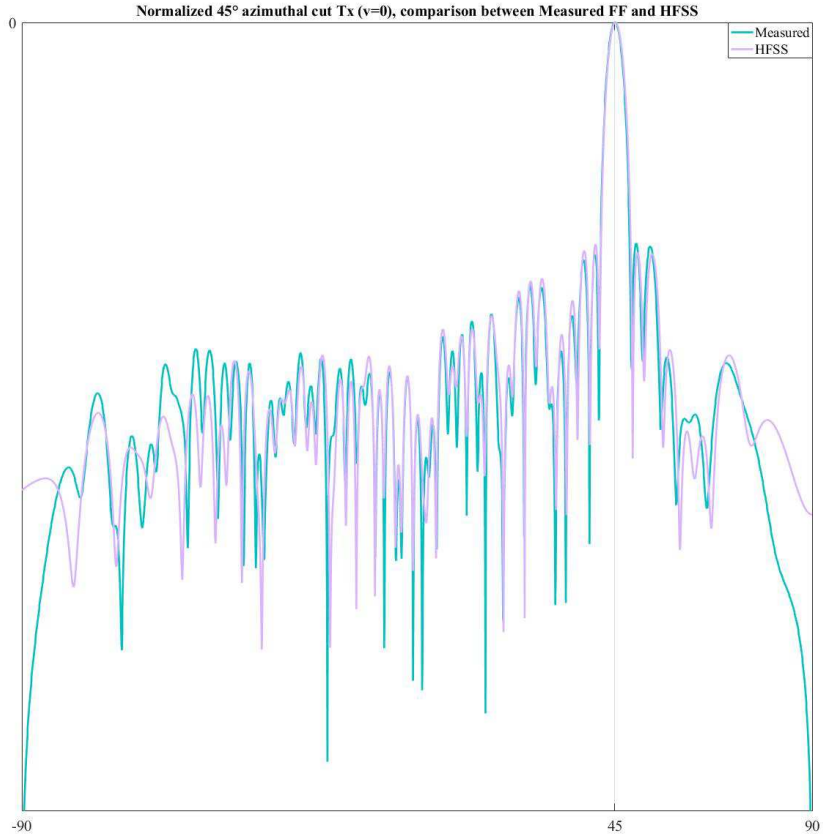

Figure 70. 45° azimuthal (v=0) cut for sparse Tx *X-Antenna*: comparison between measured and simulated (HFSS) results

5.6 Experimental results: the measured FF in Tx of the sparse X-Antenna

In this section the final results in terms of FF performances are shown as regards the Tx side. "*" have been put on the curves for the points that fall outside the mask, and details are provided in the following for what concerns their percentage with respect to the whole observation interval (→ 5.2.1). As regards the steering directions it has been possible to carry out the measurements only for 45° azimuth (v=0) and 60° of elevation (u=0), but nonetheless the antenna behavior in the cases of -45° azimuth (v=0) and -60° of elevation (u=0) is typically the same, so that it is quite likely that if the requirements are met in the former cases they are met in the latter ones as well. In Figure 74 the mask is not shown for secrecy reasons.
In particular, we have:

- the beamwidth @-3dB for the horizontal cut is LFBT-0.03°.
- The beamwidth @-3dB for the vertical cut is LFBT-0.03°.
- The beamwidth @-3dB for 45° Azimuth (v=0 cut) is LFAT+0.05°, thus it is slightly wider than the expected value. Anyway, the requirement is not met.
- The beamwidth @-3dB for 60° Elevation (u=0 cut) is LFET-0.1°.
- The rms SLL beyond ±LTF for the horizontal cut is LLRBT-6.6601dB.
- The rms SLL beyond ±LTF for the vertical cut is LLRBT-4.7762dB.
- The rms SLL beyond ±LT for 45° Azimuth (v=0 cut) is LLRAT-5.5268dB.
- The rms SLL beyond ±LT for 60° Elevation (u=0 cut) is LLRET-1.3085dB.
- The percentage of points outside the mask for the vertical cut (Figure 71) is 2.48%.
- The percentage of points outside the mask for the 60° Elevation (u=0 cut) (Figure 74) is 0.74%.

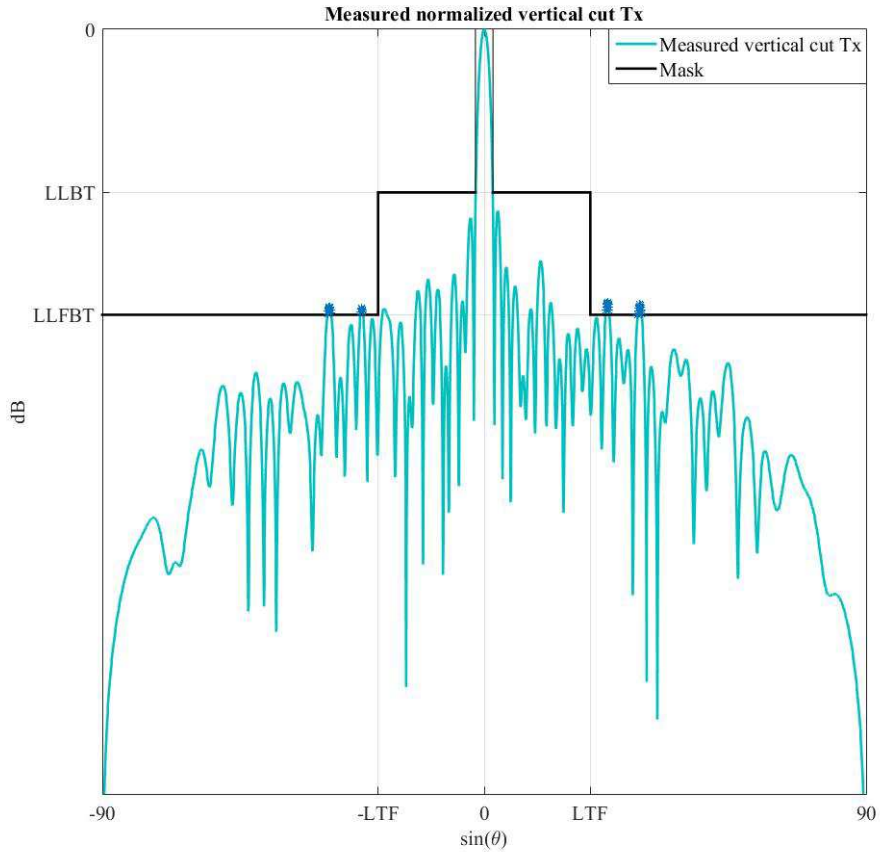

Figure 71. Vertical cut for sparse Tx *X-Antenna* at broadside

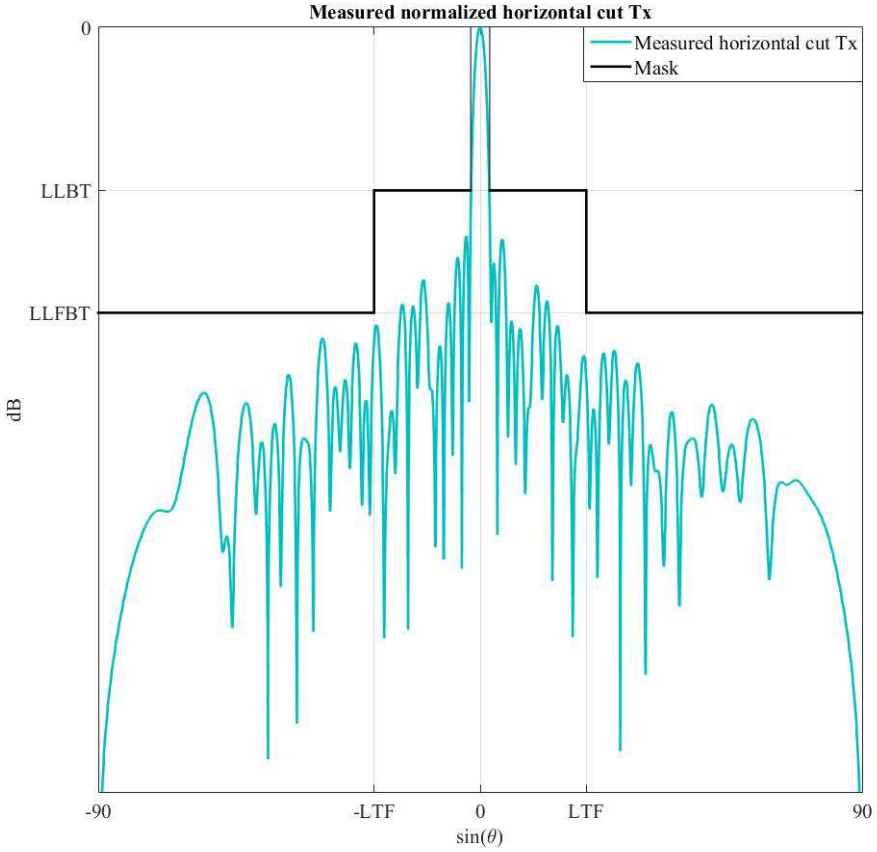

Figure 72. Horizontal cut for sparse Tx *X-Antenna* at broadside

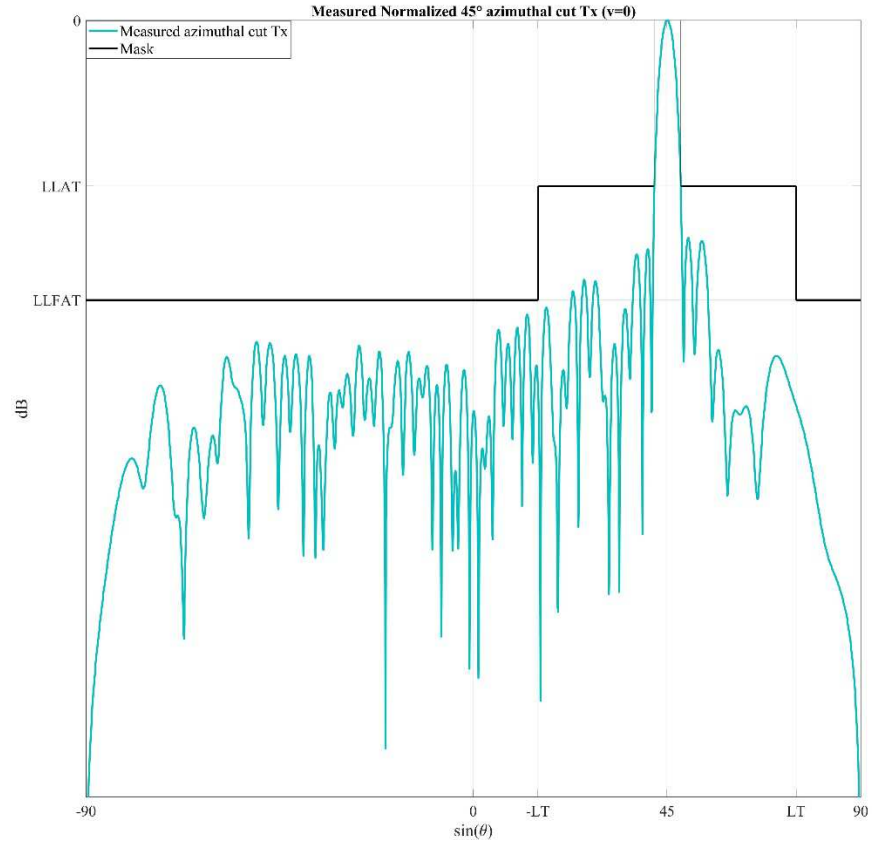

Figure 73. 45° azimuthal (v=0) cut for sparse Tx *X-Antenna*

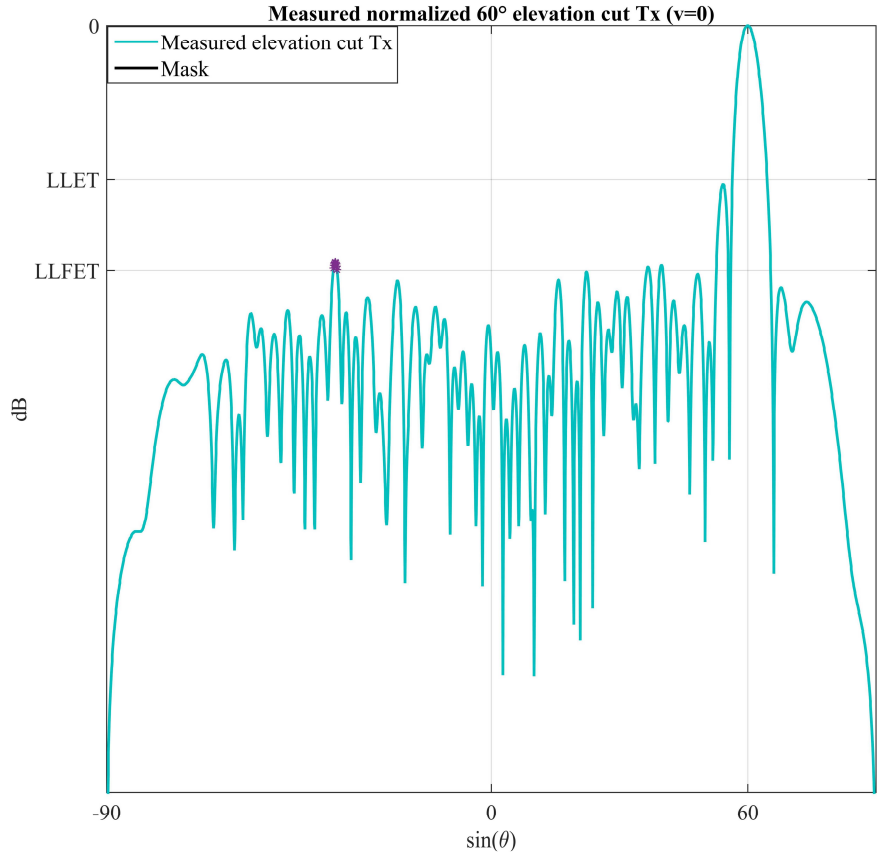

Figure 74. 60° elevation (u=0) cut for sparse Tx *X-Antenna*

5.7 Experimental results: the measured FF in Rx of the sparse X-Antenna

In this section the final results in terms of FF performances are shown as regards the Rx side. "*" have been put on the curves for the points that fall outside mask, and details are provided in the following for what concerns their percentage with respect to the whole observation interval (→ 5.2.2). As regards the steering directions it has been possible to carry out the measurements only for 45° azimuth (v=0) and 60° of elevation (u=0), but nonetheless the antenna behavior in the cases of -45° azimuth (v=0) and -60° of elevation (u=0) is typically the same, so that it is quite likely that if the requirements are met in the former cases they are met in the latter ones as well.
In particular, we have:

- the beamwidth @-3dB for the horizontal cut is LFBT-0.03°.
- The beamwidth @-3dB for the vertical cut is LFBT-0.035°.
- The beamwidth @-3dB for 45° Azimuth (v=0 cut) is LFAT-0.2361.
- The beamwidth @-3dB for 60° Elevation (u=0 cut) is LFET-0.544°.
- The rms SLL beyond ±LTF for the horizontal cut is LLRBT-5.6142dB.
- The rms SLL beyond ±LTF for the vertical cut is LLRBT-6.0223dB.
- The rms SLL beyond ±LT for 45° Azimuth (v=0 cut) is LLRAT-6.4935dB.
- The rms SLL beyond ±LT for 60° Elevation (u=0 cut) is LLRET-2.567dB.
- The percentage of points outside the mask for the vertical cut (Figure 75) is 9.899%.
- The percentage of points outside the mask for the horizontal cut (Figure 76) is 2.8534%.
- The percentage of points outside the mask for the 45° azimuthal (v=0 cut) (Figure 77) is 1.8741%.
- The percentage of points outside the mask for the 60° Elevation (u=0 cut) (Figure 78) is 17.22%, thus the requirement is not met.
- The 60° Elevation (u=0 cut) has points that are greater than LLE (see red ellipse in Figure 78), thus the requirement is not met. However, if we consider this cut for the full, standard case, we

see there is no significant difference between the two curves, see Figure 79. In particular, in such figure we highlighted the points outside the mask (whose percentage is 18.87%) on the full-antenna curve with green "*". Moreover, the main lobe is even wider than the one pertaining to the sparse case, therefore not meeting the requirements in this case could be due to the full-antenna conditions in Rx during the measurements, rather than to a synthesis issue.

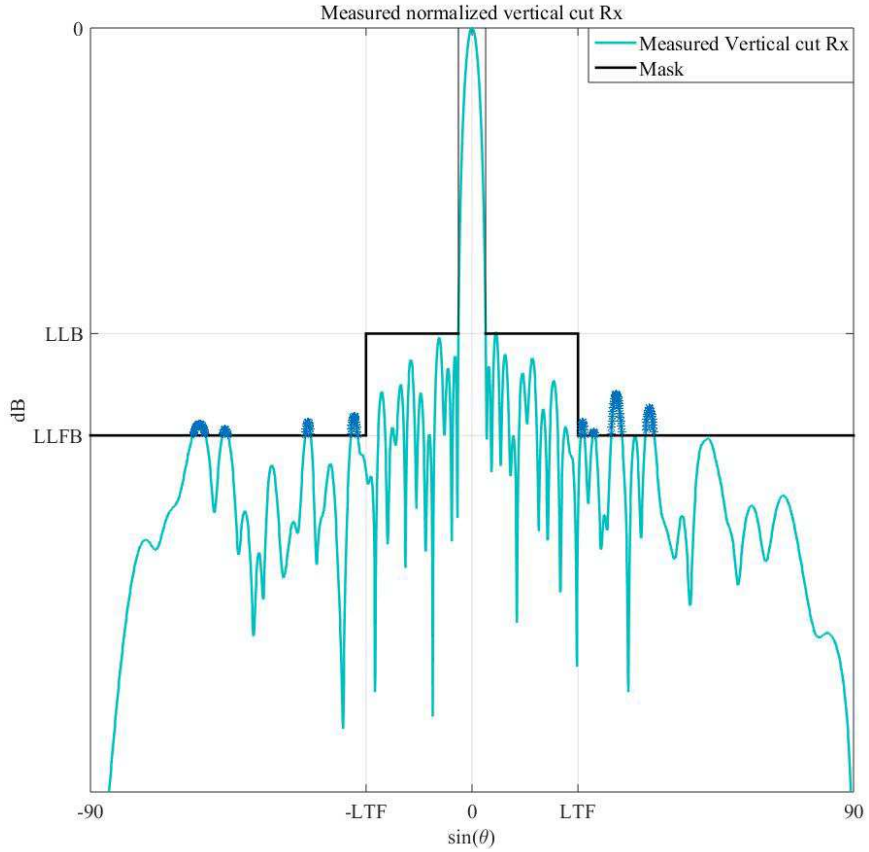

Figure 75. Vertical cut for sparse Rx *X-Antenna* at broadside

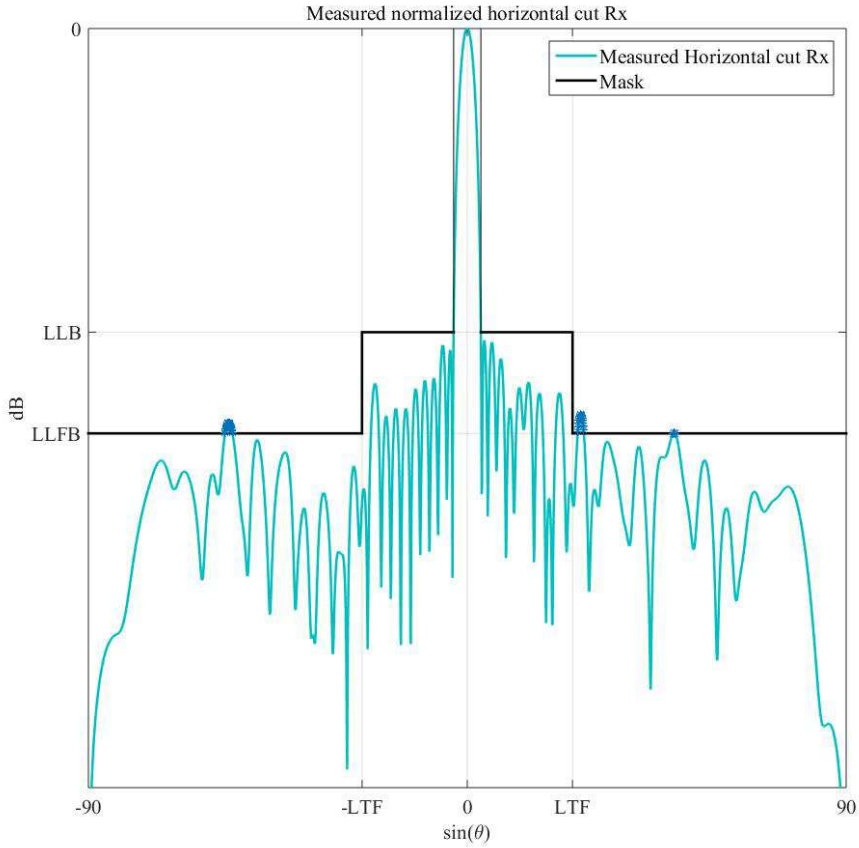

Figure 76. Horizontal cut for sparse Rx *X-Antenna* at broadside

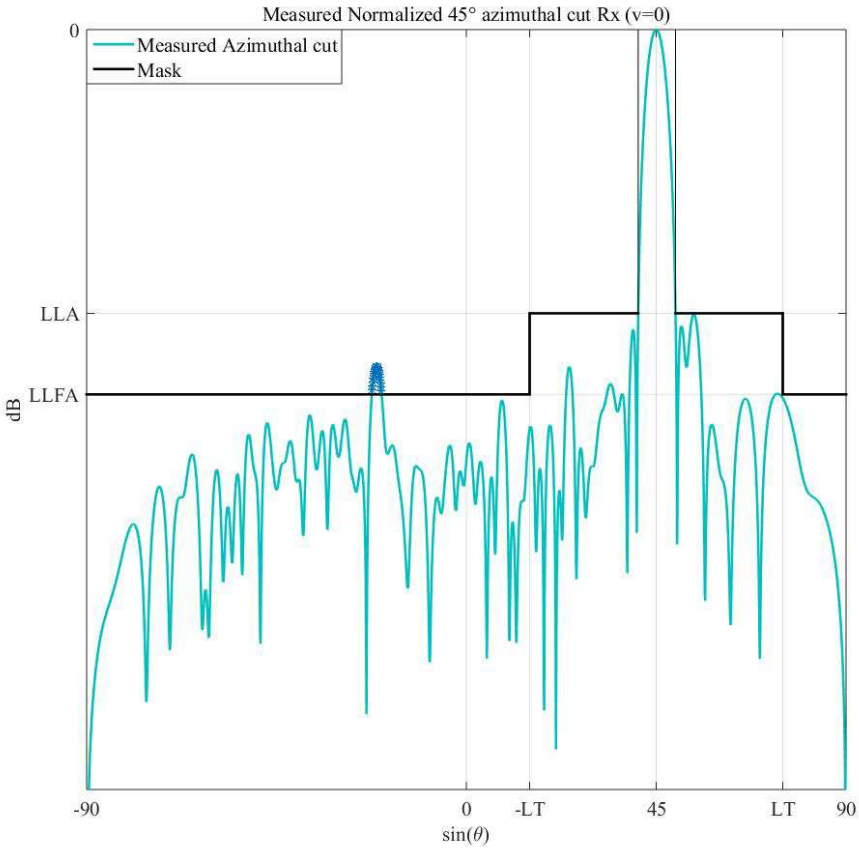

Figure 77. 45° azimuthal (v=0) cut for sparse Rx *X-Antenna*

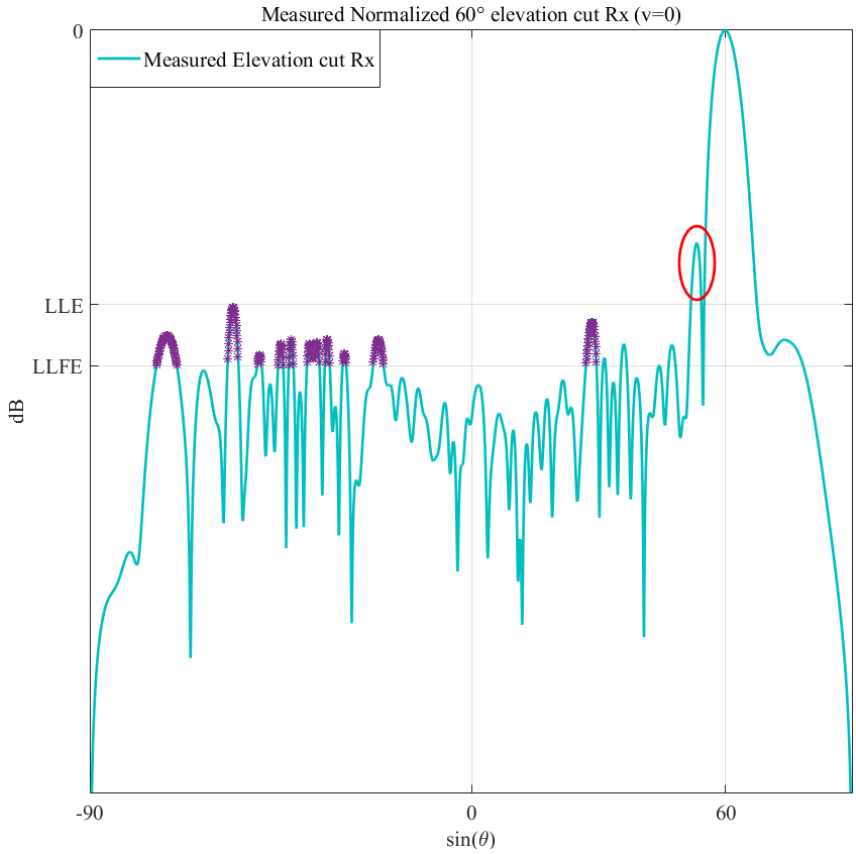

Figure 78. 60° elevation (u=0) cut for sparse Rx *X-Antenna*

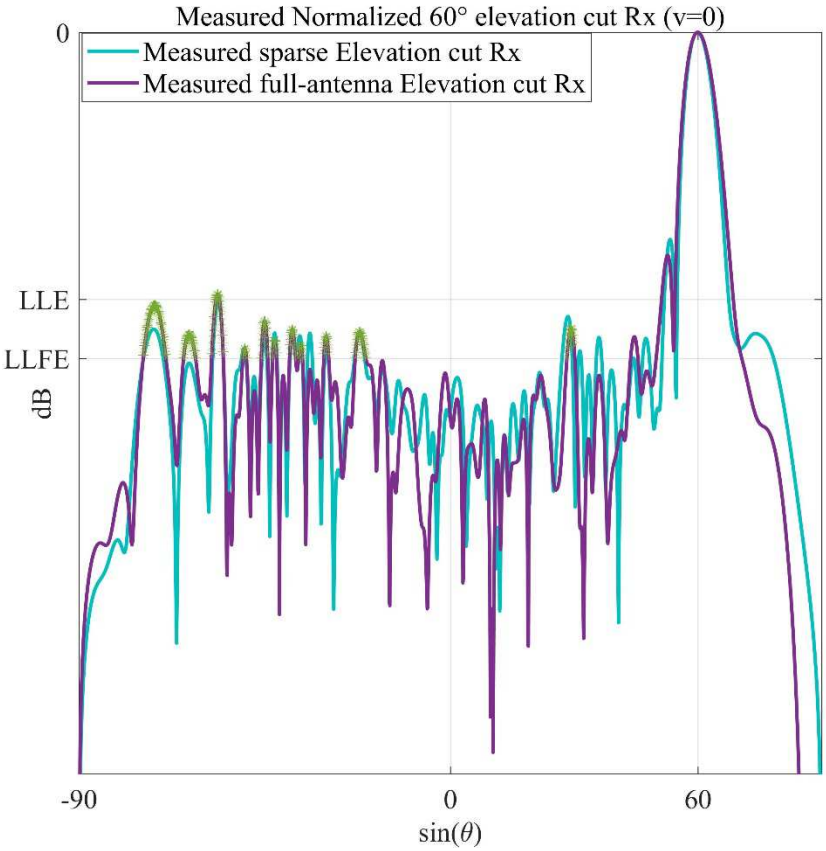

Figure 79. 60° elevation (u=0) cut for sparse and full Rx *X-Antenna*

Chapter 6

Conclusions and future research

In the present thesis work the analysis, design and application of algorithms for the synthesis of maximally sparse, planar, non-superdirective, multi-pattern and steerable arrays has been investigated with a special focus on the Tx (isophoric) case. Starting from a dense sampled, planar aperture, such algorithms allow to determine a single layout for Tx and several Rx sets of weights via convex optimizations by minimizing the number of elements. In particular, the necessity of synthesizing an isophoric array, among the other goals, lays the foundations of MIP, since it requires the introduction of binary variables (which account for presence or absence of elements for each point of the grid) in the problem formulation. The objective function becomes therefore non-convex, requiring to solve, in principle, an NP-hard problem of exponential cardinality. At the same time however, not only it allows to determine the Tx array from the synthesis process, but it also allows to apply a constraint on the minimum inter-element distance exactly, which in turn assures the sparse array to be non-superdirective. The joint Tx/Rx synthesis, together with the latter constraint are two among the most innovative contributes of this thesis. Two approaches have been presented and described to face the non-convexity of the objective function: the reweighted ℓ_1-norm and algorithms of B&B type. From a software implementation point of view the use of MIP is quite appealing because with commercial solvers like Gurobi and Mosek the user has the possibility of writing the mathematical problem exactly as in (2.2), even if the objective function is ultimately non-convex. Or, equivalently, it becomes convex in terms of binary variables (in fact it is just their sum), and yet this is transparent to the user: the solver automatically calls into play the B&B algorithms. Advantages and disadvantages clearly affect these two approaches: the reweighted ℓ_1-norm favors simulation time at the detriment of non-superdirectivity constraint, which in this case is applied just approximately. Moreover, when a Tx synthesis is strictly required, as for sparsifying the *X-antenna*, it generally returns worse performances than the case of MIP. This is due to the fact that the reweighted ℓ_1-norm enhances the gap between "zero" and "non zero" entries in the vector of interest (and according to this view it is very well suited for approximating the cost function of problem (2.2)), privileging sparsity and pushing as close to zero as possible the formers, but does not intrinsically address the heterogeneity of amplitude within the latters. After deriving the MIP solution for the *X-Antenna*, several attempts were made by using the reweighted ℓ_1-norm with the goal of obtaining a solution that could get as close to it as possible, but we did not succeed: it returned a greater number of (non-zero) elements. Apart from this, since we already had a MIP solution available we knew in advance how to set the parameter *b* in (2.23), but this is apparently not the ordinary case: normally one has to perform the optimization (2.23) for numerous values of *b*, which can easily lead to a longer overall simulation time than in the case of MIP.
On the contrary, the MIP approach privileges exactness, sparsity, global minimum and (intrinsically) Tx solutions at the detriment of time required to converge. Hence, the time budget one has available and the necessity of obtaining a Tx synthesis act as the main actors for choosing between the two solutions presented. However, it must be underlined that the design and realization process of a new AESA may take years, and in this long interval the synthesis process requests a negligible time, even if it had to take one month for instance. Furthermore, it is a process that typically must be carried out only once.

As regards the efficacy of the solutions proposed to problem (2.2), it has been shown:

- in the application of the proposed optimization schemes to benchmarks present in literature,
- in the creation of new benchmarks, and especially
- in the application to an existing antenna (the *X-antenna*) composed of thousands of elements,

whose FF performances exhibit a very good coherence with the ones predicted by the simulations carried out in HFSS, thus validating both the simulation model and the sparsification process itself and proving the antenna meets the requirements with an ENRR of 55.74%. The measurements of the sparse version of the *X-Antenna* and the relative FF performances represent another major contribute in terms of innovation in the present work.
Starting from the latest results obtained from the measurements in anechoic chamber the next steps consist in addressing the peak directivity loss problem. In fact, in sparsifying the *X-Antenna* we allowed the

solver to minimize the number of elements without any constraint on this quantity, but in order for the developed optimization schemes to be more attractive for future designs, ENRR and (peak) directivity must be put on the balance: by dropping the former one can push the latter higher. The loss in peak directivity that can be tolerated depends on higher level requirements, which are eventually related to the radar's range coverage, PD and PFA. Furthermore the sparse synthesis has been done only for the center frequency. It is supposed to be carried out for at least f_{min} and f_{max} as well. On another side the business aspect shall be considered accordingly as well, since, besides ENRR, peak directivity and other figures of performances, it is mandatory to figure out how much money saving can be earned in the realization of a sparse antenna. Anyway, such aspect may be taken into account a priori in the synthesis process by appropriately finding, more in general, the relation between (monetary) costs and (some) indexes of performances.

Bibliography

[1] G. Toso, C. Mangenot, A. G. Roederer, "Sparse and thinned arrays for multiple beam satellite applications," in *Antennas and propagation, EUCAP 2007*, pp. 1–4, 11-16 November 2007.

[2] G. Caille, Y. Cailloce, C. Guiraud, D. Auroux, T. Touya, M. Masmousdi, "Large Multibeam Array Antennas with reduced Number of active chains," in *Proc. of the Second European Conference on Ant. and Propag.*, Edinburgh, UK, November 2007.

[3] O. M. Bucci, M. D'Urso, T. Isernia, P. Angeletti, G. Toso, "Deterministic Synthesis of Uniform Amplitude Sparse Arrays via New Flexible Density Taper Techniques," *IEEE Trans. Antennas Propag.*, 58, 6, pp. 1949-1958, June 2010.

[4] B. P. Kumar, G. R. Branner, "Design of unequally spaced arrays for performance improvement," *IEEE Trans. Antennas Propag.*, vol. 47, pp. 511-523, Mar. 1999.

[5] Randy L. Haupt, "Thinned Arrays Using Genetic Algorithms", *IEEE Trans. Antennas Propag.*,vol. 42, no. 7, July 1994.

[6] L. Cen, W. Ser, Z. L. Yu, S. Rahardja, W. Cen, "Linear Sparse Array Synthesis With Minimum Number of Sensors", *IEEE Trans. Antennas Propag*., vol. 58, no. 3, pp.720-726, March 2010. [IGA]

[7] Arnon Sakonkanapong, Sarayoot Todnatee, and Chuwong Phongcharoenpanich, "Multi-Parents Genetic Algorithm Optimization Scheme for Linear Array Antenna Synthesis", *2017 IEEE International Conference on Computational Electromagnetics (ICCEM)*, 8-10 March 2017 [IGA]

[8] Yi Jiang, Shu Zhang, Qiang Guo, Xiaoming Luan, "A Hybrid Strategy Based on Weighting Density and Genetic Algorithm for the Synthesis of Uniformly Weighted Concentric Ring Arrays", *IEEE Antennas and Wireless Propagation Letters, Vol.16*, pp. 186-189, 12 May 2016 [GA]

[9] Mahmud Esad Yiğit, Tayfun Günel, "A new hybrid optimization approach based on Convex-Genetic-Taguchi algorithms for the pattern synthesis of linear antenna array", *2016 National Conference on Electrical, Electronics and Biomedical Engineering (ELECO)*, 1-3 Dec. 2016 [GA]

[10] B. V. Ha, M. Mussetta, P. Pirinoli, R. E. Zich, "Modified Compact Genetic Algorithm for Thinned Array Synthesis", *IEEE Antennas and Wireless Propagation Letters, Vol.15*, pp. 1105-1108, 27 Oct 2015 [GA]

[11] Smita Banerjee, Ved Vyas Dwivedi, "Linear Antenna Array Synthesis to Reduce the Interference in the Side Lobe Using Continuous Genetic Algorithm", *2015 Fifth International Conference on Advances in Computing and Communications (ICACC)*, 2-4 Sept. 2015 [GA]

[12] Amina Khalid, Shahzad Amin Sheikh, Ijteba Ul Hasnain Shah, Qasim Umar Khan, "Synthesis of linear antenna array using genetic algorithm to reduce peak sidelobe level", *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)*, 26-28 Nov. 2015 [GA]

[13] Parthasarathi Pal, Durbadal Mondal, "Adaptive antenna array pattern synthesis for suppressed sidelobe level and controlled null using genetic algorithm", *2014 International Conference for Convergence of Technology (I2CT)*, 6-8 April 2014 [GA]

[14] Hassan M. Elkamchouchi, Mohamed M. Hassan, "Array pattern synthesis approach using a genetic algorithm", *IET Microwaves, Antennas & Propagation, Vol. 8, pp. 1236-1240* [GA]

[15] Durbadal Mandal, Y. Naren Tapaswi, "Radiation pattern synthesis of linear antenna arrays by amplitude tapering using Genetic Algorithm", *Applied Electromagnetics Conference (AEMC)*, 18-22 Dec. 2011 [GA]

[16] Cheol-Min Seong, Myung-seok Kang, Cheol-Soo Lee, Dong-Chul Park, “Conformal array pattern synthesis on a curved surface with quadratic function using adaptive genetic algorithm”, *Microwave Conference Proceedings (APMC),* 5-8 Nov. 2013 [GA]

[17] M. D’Urso and T. Isernia, ”Solving some array synthesis by means of an effective hybrid approach”, *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 750-759, March 2007. [sia SA che GA]

[18] T. Isernia, F. J. Ares Pena, O. M. Bucci, M. D’Urso, J. Fondevila Gómez, and J. A. Rodríguez, “A Hybrid Approach for the Optimal Synthesis of Pencil Beams Through Array Antennas”, *IEEE Trans. Antennas Propag.*, vol. 52, no. 11, pp. 2912 – 2918, November 2004. [SA]

[19] W. Tang, Y. Zhou, ”Frequency invariant power pattern synthesis for arbitrary arrays via simulated annealing”, *Electronics Letters,* Vol. 46, Issue: 25, December 9 2010, pp. 1647-1648. [SA]

[20] N.H. Farhat, B. Bai, ”Phased-array antenna pattern synthesis by simulated annealing”, *Proceedings of the IEEE,* Vol. 75, Issue: 6, June 1987, pp. 842-844. [SA]

[21] V. Murino, A. Trucco, C.S. Regazzoni, ”Synthesis of unequally spaced arrays by simulated annealing”, *IEEE Transactions on Signal Processing,* Vol. 44, Issue: 1, Jan 1996, pp. 119-122 [SA]

[22] J.A. Ferreira, F. Ares, ” Pattern synthesis of conformal arrays by the simulated annealing technique”, *Electronics Letters*, Vol. 33, Issue 14, 3 Jul 1997, pp. 1187-1189 [SA]

[23] Michele D'Urso, Alessio Iacono, Antonio Iodice, Giorgio Franceschetti, ”Optimizing uniformly excited time-modulated linear arrays”, *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP),* 11-15 April 2011 [SA]

[24] L. Kurup, M. Himdi, A. Rydberg, “Synthesis of Uniform Amplitude Unequally Spaced Antenna Arrays using the Differential Evolution Algorithm”, *IEEE Trans. Antennas Propag.*,vol. 51, no. 9, pp. 2210-2217, September 2003.

[25] Ananya Mukherjee, Sujit Kumar Mandal, Rowdra Ghatak, “Synthesis of non-uniformly spaced planar array geometry using Differential Evolution algorithm”, *2016 IEEE Indian Antenna Week.*, 6-10 June 2016 [DE]

[26] Heng Liu, Hongwei Zhao, Weimei Li, Bo Liu, “Synthesis of Sparse Planar Arrays Using Matrix Mapping and Differential Evolution”, *IEEE Antennas and Wireless Propagation Letters,* Vol. 15, pp. 1905-1908, Mar 2016 [DE]

[27] Xin Li, Wen Tao Li, Xiao Wei Shi, Xia Ai, “An improved multi-objective differential evolution algorithm and its application for array pattern synthesis”, *2012 International Conference on Microwave and Millimeter Wave Technology (ICMMT),* July 2012 [DE]

[28] Ananda Kumar Behera, Aamir Ahmad, S. K. Mandal, G. K. Mahanti, Rowdra Ghatak, “Synthesis of cosecant squared pattern in linear antenna arrays using differential evolution”, *2013 IEEE Conference on Information & Communication Technologies*, Apr 2013 [DE]

[29] Fenggan Zhang, Weimin Jia, Minli Yao, “Linear Aperiodic Array Synthesis Using Differential Evolution Algorithm”, *IEEE Antennas and Wireless Propagation Letters*, Vol. 12, pp. 797-800, Jun 2013 [DE]

[30] Xuzhen Zhang, Pingui Jia, “Unequally spaced arrays synthesis based on modified differential evolution”, *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet),* Apr 2012 [DE]

[31] Sotirios K. Goudos, Katherine Siakavara, Theodoros Samaras, Elias E. Vafiadis, John N. Sahalos, “Sparse Linear Array Synthesis With Multiple Constraints Using Differential Evolution With Strategy Adaptation”, *IEEE Antennas and Wireless Propagation Letters*, Vol. 10, pp. 670-673, Jul 2011 [DE]

[32] M. Clerc and J. Kennedy, “The particle swarm: Explosion, stability, and convergence in a multidimensional complex space,” *IEEE Trans. Evo. Comp.*, vol. 6, pp. 58–73, 2002.

[33] Robinson and Y. Rahmat-Samii, “Particle swarm optimization in electromagnetics,” *IEEE Trans. Antennas Propag.*, vol. 52, no. 2, pp.397–407, Feb. 2004.

[34] D. Gies and Y. Rahmat-Samii, “Particle swarm optimization for reconfigurable phase-differentiated array design”, *Microw. Opt. Technol.Lett.*, Aug. 2003.

[35] D. Mandal, A.K. Bhattacharjee, S.P. Ghoshal, “Linear Antenna Array Synthesis Using Improved Particle Swarm Optimization” *2011 Second International Conference on Emerging Applications of Information Technology (EAIT)*, 19-20 Feb. 2011 [PSO]

[36] Durbadal Mandal, Sudipta Das, Somen Bhattacharjee, Anup Bhattacharjee, Sakti Ghoshal, “Linear antenna array synthesis using Novel Particle Swarm Optimization”, *2010 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, 3-5 Oct. 2010. [PSO]

[37] M.M. Khodier, C.G. Christodoulou, “Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization”, *IEEE Transactions on Antennas and Propagation*, Vol. 53, Issue 8, pp. 2674–2679, Aug. 2005. [PSO]

[38] Edson R. Schlosser, Sabrina M. Tolfo, Marcos V. T. Heckler, “Particle Swarm Optimization for antenna arrays synthesis”, *2015 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*, Nov. 2015. [PSO]

[39] Shoulei Ma, Hailin Li, Aihua Cao, Jing Tan, Jianjiang Zhou, “Pattern synthesis of the distributed array based on the hybrid algorithm of particle swarm optimization and convex optimization”, *2015 11th International Conference on Natural Computation (ICNC)*, Jan. 2016. [PSO]

[40] Y. Liu, Z. Nie, Q. H. Liu, “Reducing the Number of Elements in a Linear Antenna Array by the Matrix Pencil Method,” *IEEE Trans. Antennas Propag.*, vol. 56, no 9, pp. 2955-2962 September 2008.

[41] Y. Liu, Z. Nie, Q. H. Liu, “Reducing the Number of Elements in the Synthesis of Shaped-Beam Patterns by the Forward-Backward Matrix Pencil Method,” *IEEE Trans. Antennas Propag.*,vol 58, no.2, pp. 604-608, January 2010.

[42] Massa, P. Rocca, G. Oliveri, “Compressive sensing in electromagnetics – A review”, *IEEE Antennas and Propagation magazine*, pp. 224-238, Vol. 57, no. 1, Feb. 2015.

[43] Andrea F. Morabito, Antonia R. Laganà, Tommaso Isernia, “Improving Performances of Compressive Sensing in the Synthesis of Maximally-Sparse Arrays”, *2014 IEEE Antennas and Propagation Society International Symposium (APSURSI)*, 6-11 July 2014.

[44] F. Viani, G. Oliveri, and A. Massa, “Compressive Sensing Pattern Matching Techniques for Synthesizing Planar Sparse Arrays”, *IEEE Trans. Antennas Propag.*, Vol. 61, no. 9, September 2013.

[45] G. Oliveri and A. Massa, "Bayesian compressive sampling for pattern synthesis with maximally sparse non-uniform linear arrays", *IEEE Trans. Antennas Propag.*, vol. 59, pp. 467-481, Feb. 2011.

[46] B. Fuchs, “ Synthesis of sparse arrays with focused or shaped beampattern via sequential convex optimizations, ” *IEEE Trans. Antennas Propag.*, vol. 60, no. 7, pp. 3499 – 3503, Jul. 2012.

[47] G. Prisco and M. D’Urso, “Maximally sparse arrays via sequential convex optimizations” *IEEE Antennas Wireless Propag. Lett.*, vol. 11, pp. 192 – 195, Feb. 2012.

[48] G. Prisco, M. D’Urso, Roberto M. Tumolo, “Maximally Sparse, steerable and Nonsuperdirective Array antennas via Convex Optimizations”, *IEEE Trans. on Ant. and Propag.*, Vol. 64, Issue 9, pp. 3840–3849, September 2016.

[49] Siew Eng Nai, Wee Ser, Zhu Liang Yu, Huawei Chen, "Beampattern Synthesis for Linear and Planar Arrays With Antenna Selection by Convex Optimization", *IEEE Trans. Antennas Propag.*, Vol. 58, Issue 12, pp. 3923-3930, Dec. 2010.

[50] Yanhui Liu, Pengfei You, Chunhui Zhu, Xiaofeng Tan, and Qing Huo Liu, "Synthesis of Sparse or Thinned Linear and Planar Arrays Generating Reconfigurable Multiple Real Patterns by Iterative Linear Programming", *Progress In Electromagnetics Research*, Vol. 155, 27–38, 2016.

[51] C. Hansen, "Phased Array Antennas", 1998 by John Wiley & Sons, Inc.

[52] Matthew B. Hawes, Wei Liu, "Compressive sensing-based approach to the design of linear robust sparse antenna arrays with physical size constraint", *IET Microwaves, Antennas & Propagation*, Feb. 2014

[53] Matthew B. Hawes, Wei Liu, "Pattern synthesis of linear antenna arrays using a genetic algorithm with physical size constraint", *2012 6th European Conference on Antennas and Propagation (EUCAP)*, 26-30 March 2012

[54] Matthew B. Hawes, Wei Liu, Richard J. Langley, "Sparse antenna array design via compressive sensing", *IET Colloquium on Antennas, Wireless and Electromagnetics 2014*, 27-27 May 2014

[55] Stephen Boyd, Lieven Vandenberghe, "Convex Optimization", Cambridge University Press 2004

[56] Jon Dattorro, "Convex Optimization & Euclidean distance geometry", Mεβoo Publishing USA, 2005

[57] Der-San Chen, Robert G. Batson, Yu Dang, "Applied Integer Programming – Modeling and solution", John Wiley & Sons, 2010

[58] Rajesh Kumar Arora, "Optimization – Algorithms and Applications", CRC Press Taylor and Francis Group, 2015

[59] Michael Junger, Thomas Liebling, Denis Naddef, George Nemhauser, William Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, "50 Years of integer programming 1958-2008. From early years to ther State-of-the-art", Springer-Verlag Editor, 2010

[60] R.E. Gomory, "Outline of an algorithm for integer solutions to linear programs", *Bulletin of the American Mathematical Society* 64 (1958) 275-278.

[61] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem", *Operations Research 2* (1954) 393–410.

[62] P.M. Narendra and K. Fukunaga, "A Branch and Bound Algorithm for Feature Subset Selection," *IEEE Trans. Computers*, vol. 26, no. 9, pp. 917-922, Sept. 1977.

[63] Petr Somol, Pavel Pudil, "Fast Branch & Bound Algorithms for Optimal Feature Selection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 26, no.7, pp. 900 – 912, July 2004.

[64] Robert E. Bixby, "A brief history of linear and Mixed Integer Programming computation", *Documenta Math*., 2010

[65] A. H. Land, A. G. Doig, "An Automatic Method of Solving Discrete Programming Problems" *Econometrica*, Vol. 28, No. 3. (Jul., 1960), pp. 497-520.

[66] Daniel Scholz, "Deterministic Global Optimization - Geometric Branch-and-bound Methods and Their Applications", Springer New York Dordrecht Heidelberg London, 2012

[67] K. Fukunaga, "Introduction to Statistical Pattern Recognition", second ed. Academic Press, Inc., 1990.

[68] Y. Hamamoto, S. Uchimura, Y. Matsuura, T. Kanaoka, and S.Tomita, "Evaluation of the Branch and Bound Algorithm for Feature Selection," *Pattern Recognition Letters*, vol. 11, no. 7, pp. 453-456, July 1990.

[69] Alexandros C. Dimopoulos, Christos Pavlatos, George Papakonstantinou, "A General Purpose Branch and Bound Parallel Algorithm", *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 17-19 Feb. 2016

[70] Kimmo Nieminenm, Sampo Ruuth, Istvan Maros, "Genetic algorithm for finding a good first integer solution for MILP", April 2003

[71] M. E Cardoso, R. L. Salcedo, S. Feyo de Azevedo and D. Barbosa, "A simulated annealing approach to the solution of minlp problems", *Pergamon, Computers chem.* Engng Vol. 21. No. 12. pp. 1349-1364. 1997.

[72] David Abramson, Marcus Randall, "A simulated annealing code for General Integer Programs", *Annals of Operations Research* January 1999, Volume 86, Issue 0, pp 3-21

[73] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization", *J. Fourier Anal. Appl.*, vol. 14, no. 5/6, pp. 877 – 905, Oct. 2008.

[74] C. Bencivenni, M. V. Ivashina, R. Maaskant, J. Wettergren, "Design of Maximally Sparse Antenna Arrays in the Presence of Mutual Coupling", *IEEE Antennas and Wireless Propagation Letters*, Vol.14, pp. 159-162, September 2014.

[75] C. Bencivenni, M. V. Ivashina, R. Maaskant, J. Wettergren, "Synthesis of Maximally Sparse Arrays Using Compressive Sensing and Full-Wave Analysis for Global Earth Coverage Applications", *IEEE Transactions on Antennas and Propagation*, Vol. 64, Issue: 11, Nov. 2016.

[76] C. Bencivenni Xiaowen Zhao, Qingshan Yang, Yunhua Zhang, "Compressed sensing approach for pattern synthesis of maximally sparse non-uniform linear array", *IET Microwaves, Antennas & Propagation*, Vol. 8, Issue 5, April 2014

[77] Ivan W. Selesnick ; Ilker Bayram, "Sparse Signal Estimation by Maximally Sparse Convex Optimization", *IEEE Transactions on Signal Processing*, Vol. 62, Issue 5, March 2014

[78] Sverre Holm, Bjørnar Elgetun, Geir Dahl, "Weight- and Layout-Optimized Sparse Arrays", *Proc. Int. Workshop On Sampling Theory and Applications*, June, 1997

[79] O. M. Bucci, M. D'Urso, "Power Pattern Synthesis of Given Sources Exploiting Array Methods", *Proc. of the Second European Conference on Ant. and Propag.*, EuCAP 2007, November 2007.

[80] H. L. Van Trees, "Optimum Array Processing, Part IV of Detection, Estimation and Modulation Theory", New York: Wiley, 2002.

[81] Nai, S. E., W. Ser, Z. L. Yu, et al., "Beam pattern synthesis for linear and planar arrays with antenna selection by convex optimization," *IEEE Trans. Antennas Propag.*, Vol. 58, No. 12, 3923–3930, 2010.

[82] Robert J. Mailloux, "Array antenna handbook " (second edition) *Artech House Boston|London*, 2005

[83] T. Isernia, O. M. Bucci, and N. Fiorentino, "Shaped beam antenna synthesis problems: feasibility criteria and new strategies", *Journal of Electromagnetic Waves and Applications*, Vol. 12, 103-138, 1998

[84] Bucci, O. M., and G. Franceschetti, ""On the spatial bandwidth of scattered fields", *IEEE Trans. on Antennas and Propagation*, Vol. 35, pp. 1445-1455, 1987.

[85] Bucci, O. M., C. Gennarelli, and C. Savarese, "Optimal interpolation of radiated fields over a sphere", *IEEE Trans. on Antennas and Propagation*, Vol. 39, pp. 1633-1643, 1991.

[86] Fletcher, R., "Practical methods of optimization", *New York, Wiley*, 1990.

[87] Bucci, O. M., G. Franceschetti, "On the degrees of freedom of the scattered fields", *IEEE Trans. on Antennas and Propagation*, Vol. 37, No.7, July 1989.

[88] Dolph, C. L., ‘‘A Current Distribution for Broadside Arrays Which Optimizes the Relationship Between Beamwidth and Sidelobe Level,’’ *Proc. IRE*, Vol. 34, June 1946, pp. 335–345.

[89] Schelkunov, S. A., ‘‘A Mathematical Theory of Linear Arrays,’’ *Bell System Tech. J.*, 1943, pp. 80–107.

[90] Taylor, T. T., ‘‘Design of Line Source Antennas for Narrow Beamwidth and Low Sidelobes’’, *IEEE Trans. on Antennas and Propagation*, Vol. AP-3, January 1955, pp. 16–28.

[91] Van der Mass, C. J., ‘‘A Simplified Calculation for Dolph-Chebyshev Arrays’’, *J. Appl. Phys.*, Vol. 25, No. 1, pp. 121–124.

[92] Richard M. Leahy, Brian D. Jeffs, ‘‘On the design of maximally sparse arrays’’, *IEEE Trans. on Antennas and Propagation*, Vol. 39, No. 8, August 1991.

[93] A. Pietsch, ‘‘Approximation spaces’’, *J. Approx. Theory*, vol. 32, pp. 115-134, 1981.

[94] I. Barrodale and F. Roberts, ‘‘Application of mathematical programming to l_p approximation’’, *Nonlinear programming*, J. Rosen et al., Eds. New York: Academic, 1970.

[95] J. J. Fuchs and B. Delyon, ‘‘Minimal l_1 norm reconstruction function for oversampled signals: Applications to time-delay estimation’’, *IEEE Trans. Infor. Theory*, vol. 46, no. 4, pp. 1666-1673, Jul. 2000

[96] D. L. Donoho and X. Huo, ‘‘Uncertainty principles and ideal atomic decomposition’’, *IEEE Trans. Infor. Theory*, vol. 47, no. 7, pp. 2845-2862, Nov. 2001.

[97] D. L. Donoho and J. Tanner, ‘‘Counting faces of randomly-projected polytopes when the projection radically lowers dimension’’, *Journal of the American Mathematical Society*, 2006.

[98] E. J. Candes, J. Romberg and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements”, *Comm. Pure Appl. Math.*, vol.59, no. 8, pp. 1207 – 1223, Aug. 2006.

[99] R. M. Tumolo, M. D'Urso, G. Prisco, and A. Buonanno, "Fast synthesis of planar, maximally thinned arrays," *Progress In Electromagnetics Research Letters*, Vol. 68, 47-52, 2017.

[100] Y. T. Lo, S. W. Lee, Q. H. Lee, “Optimization of Directivity Ratio of an Arbitrary and Signal-to-Noise Antenna Array”, *Proceedings of the IEEE*, Vol. 54, no. 8, August, 1966.

[101] https://www.pinterest.com/pin/26106872818064201/ , https://s-media-cache-ak0.pinimg.com/originals/11/1e/07/111e0777d67954a9d29e35ca64a9f5ef.jpg

[102] Akshay Agrawala, Robin Verschuerenb, Steven Diamonda, and Stephen Boyda, “*A Rewriting System for Convex Optimization Problems*”, arXiv:1709.04494 [math.OC], September 2017.